Institut National Polytechnique de Lorraine
Centre de Recherche en Informatique de Nancy

ENSEM
Ecole Doctorale IAE + M
Département de Formation Doctorale en Informatique

LE TEMPS DANS LES ARCHITECTURES DE COMMUNICATION: APPLICATION AU RÉSEAU DE TERRAIN FIP

THESE

Soutenue publiquement le 12 juillet 1994
pour l'obtention du titre de
Docteur en Informatique de l'Institut National Polytechnique de Lorraine

par

Pascal LORENZ

Membres du jury:

J.P. Haton	(président)
J.P. Elloy	(rapporteur)
	orteur)
	teur)
	recteur)
J.P. Bardinet	(invité)

Institut National Polytechnique de Lorraine
Centre de Recherche en Informatique de Nancy

ENSEM
Ecole Doctorale IAE + M
Département de Formation Doctorale en Informatique

LE TEMPS DANS LES ARCHITECTURES DE COMMUNICATION: APPLICATION AU RÉSEAU DE TERRAIN FIP

THESE

Soutenue publiquement le 12 juillet 1994
pour l'obtention du titre de
Docteur en Informatique de l'Institut National Polytechnique de Lorraine

par

Pascal LORENZ

Membres du jury:

J.P. Haton	(président)
J.P. Elloy	(rapporteur)
G. Juanole	(rapporteur)
J.P. Thomesse	(directeur)
Z. Mammeri	(co-directeur)
J.P. Bardinet	(invité)

A ma Famille,
A mes Amis,

Remerciements

Les travaux présentés ont été effectués au Centre de Recherche en Informatique de Nancy (CRIN) dans le cadre d'une convention CIFRE avec WORLDFIP europe.

Je tiens à exprimer toute ma gratitude à Monsieur Jean-Pierre THOMESSE, mon directeur de thèse, pour l'accueil dans son équipe, pour ses précieux conseils ainsi que pour ses encouragements durant mon travail de recherche.

Je remercie également Monsieur Zoubir MAMMERI pour avoir suivi de près mes travaux. Sans ses conseils, le travail n'aurait pas pu être ce qu'il est.

Je remercie Monsieur Jean-Pierre ELLOY pour avoir accepté d'être rapporteur de ces travaux et pour ses nombreuses remarques qui ont permis d'améliorer notablement la qualité de la thèse.

Je tiens à remercier également Monsieur Guy JUANOLE pour ses commentaires qui ont conduit à l'amélioration de cette thèse et pour avoir accepté d'être rapporteur.

Monsieur Jean-Paul HATON me fait l'honneur de présider ce jury de thèse. Je l'en remercie vivement.

Je remercie Monsieur Jean-Pierre BARDINET responsable de mes travaux au sein de WORLDFIP europe.

Enfin mes remerciement vont également à tous mes collèges et amis de l'équipe de recherche "Informatique Industrielle" et de WORLDFIP europe pour leurs nombreux conseils, leur amitié et leur accueil qui m'ont permis d'accomplir ce travail de manière efficace et agréable.

Sommaire

Introduction générale

Chapitre 1: La communication dans les Systèmes Automatisés

Chapitre 2: Spécification des contraintes temporelles

Chapitre 3: Modélisation et validation des mécanismes de fenêtres temporelles

Chapitre 4: Application au réseau de terrain FIP

Conclusion générale

Bibliographie

Annexe: Caractéristiques techniques et état de l'art de la normalisation des réseaux de terrain

Table des matières

Introduction générale

Introduction générale

Le temps réel joue un grand rôle dans le domaine de la communication et tout particulièrement pour les réseaux de terrain où le respect des contraintes de temps permet d'assurer la cohérence des fonctions dans les systèmes automatisés distribués.

De plus en plus, les systèmes automatisés doivent gérer et communiquer des informations en respectant des contraintes de temps plus ou moins strictes. Ainsi, une certaine information est considérée comme valide du point de vue temporel, si elle est prise en compte ni trop tôt, ni trop tard.
Il est donc nécessaire de savoir si l'information a été reçue en respectant les contraintes de temps concernant :

- le traitement de l'information ainsi que
- la communication.

De plus, à tout instant, il doit être possible de savoir si une certaine variable a encore un sens pour l'application ou si cette dernière est obsolète.

Le but de notre étude est de proposer des mécanismes temporels à intégrer aux architectures de communication temps réel. Ceux-ci vont permettre de savoir si la communication s'est déroulée en respectant les contraintes de temps énoncées par l'utilisateur.

La thèse est organisée en quatre chapitres:

Chapitre 1: La communication dans les Systèmes Automatisés

Chapitre 2: Spécification des contraintes temporelles

Chapitre 3: Modélisation et validation des mécanismes de fenêtres temporelles

Chapitre 4: Application au réseau de terrain FIP

Nous avons choisi de décrire, dans un premier chapitre, la communication dans les Systèmes Automatisés. Ceci nous a conduit à nous intéresser aux RLI (Réseaux Locaux Industriels) et tout particulièrement aux réseaux de terrain, dans lesquels les contraintes de temps sont les plus importantes. Ensuite, en nous attachant aux aspects temporels, nous étudierons les différents modèles d'architectures fonctionnelles que l'on peut rencontrer dans les Systèmes Automatisés, ainsi que le passage des architectures fonctionnelles et matérielles vers les architectures opérationnelles. Pour terminer, nous développerons différents modèles de communication que l'on rencontre généralement dans les

architectures opérationnelles, c'est-à-dire les modèles producteur(s)/ consommateur(s) et client(s)/serveur(s).

Dans un second chapitre, nous nous attacherons aux aspects temporels dans les architectures de communication et principalement au modèle de communication producteur/consommateur. Nous introduirons alors des mécanismes temporels permettant d'indiquer si les contraintes de temps ont été respectées ou non. La logique des intervalles sera utilisée pour spécifier les contraintes de temps. Nous montrerons ensuite que l'on peut distinguer différents types de fenêtres temporelles: les fenêtres temporelles liées aux étapes de production, d'émission, de réception et de consommation de l'information, ainsi que les fenêtres temporelles liées à la validité temporelles des données transmises. Nous proposerons alors un ensemble de règles permettant de construire et de lier entre elles ces fenêtres temporelles.

Dans un troisième chapitre, nous indiquerons comment élaborer des statuts temporels permettant de connaître, en fonction des règles énoncées précédemment, l'état de la communication à un instant donné. Pour cela, nous développerons une machine d'états générale permettant de générer un ensemble de statuts temporels. Cette machine d'états sera validée à l'aide de l'outil AUTO [SIM 89] et implantée au dessus de TCP/IP. Nous montrerons ensuite que notre étude peut s'appliquer à d'autres modèles de communication, tels que les modèles de communication client(s)/serveur(s).

Dans un dernier chapitre, nous étudierons les aspects temporels du réseau de terrain FIP en montrant comment, en restreignant les fonctionnalités de la machine d'états générale introduite dans le chapitre précédent, on peut retrouver les aspects temporels présents dans le réseau de terrain FIP. De nouveaux mécanismes temporels permettant de compléter les mécanismes existants dans le réseau FIP seront ensuite proposés. Nous montrerons également comment les mécanismes actuels de FIP sont implantés dans le composant FULLFIP [CEG 92]. Enfin pour terminer, nous présenterons de nouvelles machines d'états permettant de gérer les aspects temporels des listes de variables dans le réseau FIP.

Dans l'annexe, nous présenterons les principales caractéristiques techniques de différents réseaux de terrain et nous ferons un état de l'art concernant la normalisation internationale, européenne et française des réseaux de terrain.

Chapitre 1

La communication dans les Systèmes Automatisés

Chapitre 1

La communication dans les Systèmes Automatisés

1. Introduction

Un SA (Système Automatisé) est un système dont le but est d'augmenter la valeur ajoutée d'un flux d'entrée (matières premières, informations, ...) tout en obéissant à des impératifs de rentabilité et de qualité [GRE 85]. Il est constitué de deux sous-systèmes correspondant aux flux de produits et aux flux d'informations. Ces sous-systèmes sont généralement désignés par les termes de processus physique et de SI (Système d'Information). Un SA traite donc en entrée, un ensemble de flux de produits et d'informations et génère ainsi en sortie des produits finis ou semi-finis.

Le processus physique (de production, de transport, etc...) est un système destiné à faire des opérations physiques de transformation, de stockage et/ou de transport sur des flux d'entrée de matières premières. Il est composé d'un ensemble de machines effectuant des opérations selon un procédé déterminé sur les flux d'entrée et il fournit en sortie des produits finis ou semi-finis. On peut parler de partie opérative si l'on se place dans un contexte d'automatisme.

Le SI est un système qui doit faire des opérations de traitement, de mémorisation et de communication sur les flux d'information [DEL 89]. Le SI intervient dans le contrôle et la conduite du processus de production. C'est également vers lui que sont dirigés différents types d'information concernant le fonctionnement du SA et qui en élabore un historique. Le SI est donc composé de dispositifs de traitement de l'information matérialisés par des calculateurs, des automates, des régulateurs et auxquels sont associés des moyens de mémorisation et de communication. On peut parler à ce niveau de partie commande si l'on se place dans un contexte d'automatisme.

Un SA dans son contexte peut se représenter de la manière suivante:

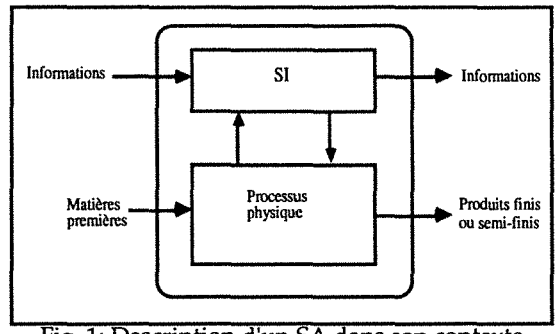


Fig. 1: Description d'un SA dans son contexte

Dans le contexte de compétitivité internationale, le CIM (Computer Integrated Manufacturing) permet aux diverses fonctions d'une entreprise d'être plus performantes en offrant la possibilité de mettre en oeuvre plus de communication, ce qui permet d'obtenir un environnement plus cohérent. Le CIM permet donc d'harmoniser les diverses fonctionnalités.

Une application de CIM se compose de fonctions telles que le contrôle, le suivi de production, la maintenance, la commande automatique et a pour but d'accroître la productivité, la flexibilité et l'évolutivité. Ainsi, on voit apparaître de nouvelles architectures intégrant, de manière modulaire, les fonctions informatisées telles que le contrôle de la qualité, la maintenance et le suivi de production qui permettent de reconsidérer la structure et l'organisation de l'entreprise afin de mettre en oeuvre une meilleure coopération entre les différentes fonctions de l'entreprise.

Divers projets sont liés à ce concept de CIM, citons le projet européen ESPRIT n°688 CIM-OSA (Computer Integrated Manufacturing-Open System Architecture) [DID 93], [KOT 93], [LAV 88], [ESP 92] qui décrit un modèle de coopération pour les différentes opérations de l'entreprise. La finalité du CIM consiste à effectuer aussi bien des tâches de manipulation de l'information, que de contrôle des machines pour toutes les différentes activités d'une usine [BRE 93].

Une architecture CIM permet la communication entre différentes entités de même niveau ainsi qu'entre des entités de niveaux adjacents. La figure suivante permet d'illustrer la communication entre différentes fonctions, telles que par exemple, les fonctions de gestion de production, d'études/développements, de gestion d'usine:

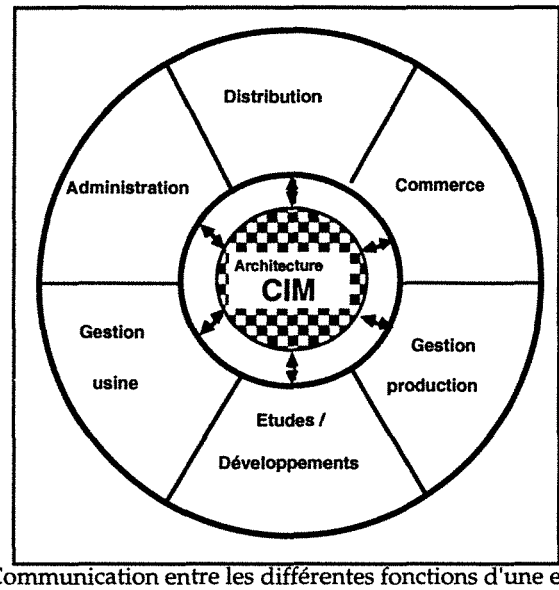


Fig. 2: Communication entre les différentes fonctions d'une entreprise

La complexité des systèmes ne peut être appréhendée que par certaines décompositions en sous-systèmes et par la définition de leurs relations ou de leur organisation. C'est ainsi que la notion d'architecture de systèmes a été introduite.
De plus, comme tout sous-système peut être informatisé, la notion d'architecture s'est traduite en termes d'architectures de systèmes informatiques autour de diverses machines et réseaux.

2. Modèles d'architectures fonctionnelles

Une architecture fonctionnelle est composée de la liste et de la structure des différentes fonctions intervenant dans une application donnée ; ainsi que des différentes relations existantes entre les fonctions. La décomposition en plusieurs niveaux de fonction fait apparaître les besoins de communications entre ces niveaux. Ceci permet donc de montrer que la communication est un concept crucial dans les SA.

A cause de la complexité des applications, on a été amené à définir des modèles hiérarchisés pour décomposer et pour raisonner par raffinement successif.

Ceci nous conduit à présenter plusieurs modèles d'architectures fonctionnelles:

- le modèle NBS qui est utilisé dans le domaine manufacturier,
- un modèle d'architecture plus spécifique au process continu,
- le modèle trois axes.

2.1 Le modèle NBS

Le modèle NBS, considéré comme un modèle d'architecture fonctionnelle pour le domaine manufacturier [BRO 87], met en évidence cinq niveaux d'abstraction: équipement, machine, cellule, atelier et usine (le niveau usine est également appelé niveau général). Ce modèle hiérarchique indique les différents points de vue que l'on peut avoir suivant le niveau où l'on se situe. Ainsi dans le modèle NBS, une machine est composée de plusieurs équipements, une cellule est composée de plusieurs machines, un atelier est composé de plusieurs cellules et une usine est composée de plusieurs ateliers [JON 86].

Une représentation possible du modèle NBS est la suivante:

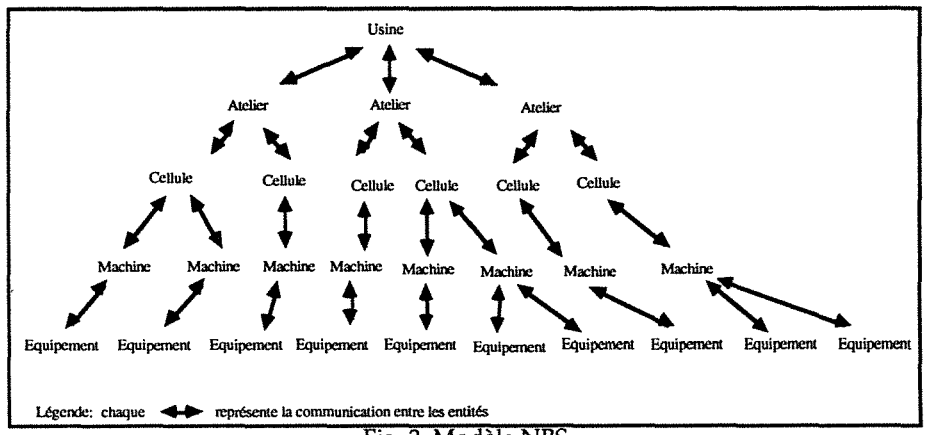


Fig. 3: Modèle NBS

Le modèle d'architecture fonctionnelle NBS privilégie le découpage du système physique en sous-systèmes sans décrire les différentes fonctions que l'on peut retrouver dans chacun des niveaux cités précédemment. Ainsi, le système de commande de la machine offre des services au niveau de la cellule en utilisant les services offerts au niveau de l'équipement.

2.2 Architecture pour les processus continus

Le modèle d'architecture fonctionnelle pour les processus continus est composé de quatre niveaux: le premier niveau concerne les capteurs/actionneurs, le second les automatismes réflexes et la régulation, le troisième la supervision et enfin le dernier niveau prend en compte les aspects optimisation et gestion [SIM 92a].

Ce modèle d'architecture pour les process continus se représente de la manière suivante:

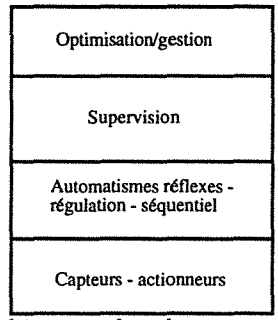


Fig. 4: Architecture dans les process continus

Ce modèle introduit différentes fonctions, mais la hiérarchisation peut prêter à confusion. Ainsi, le besoin de supervision peut être hiérarchisé et se retrouver au niveau des capteurs et des actionneurs.

Les insuffisances des deux modèles d'architecture précédents conduisent à introduire un nouveau modèle d'architecture: le modèle trois axes.

2.3 Modèle trois axes

Le modèle trois axes permet d'introduire les différentes fonctions d'une architecture CIM tout en conservant l'aspect hiérarchique [VER 89], [DEL 89], [LEC 93]. Sur un premier axe horizontal, on représente le processus physique, sur un second axe vertical les différents niveaux hiérarchiques et enfin sur le troisième axe, on place les différentes fonctions d'une application CIM.

Ce modèle trois axes se schématise de la manière suivante:

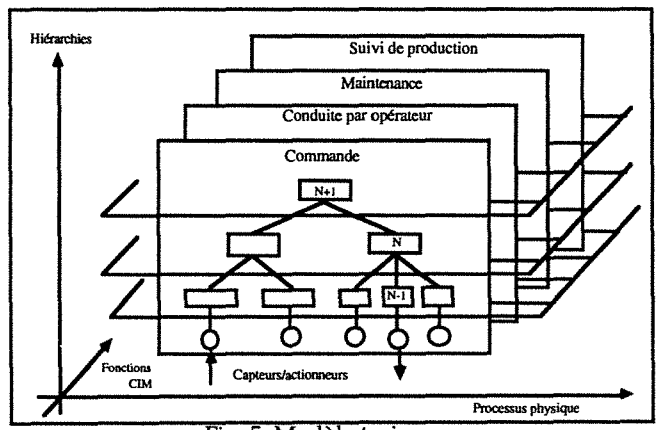


Fig. 5: Modèle trois axes

Dans le modèle trois axes, l'axe horizontal modélise, par son interfaçage avec le système de commande, le processus physique. Le processus physique peut alors être décomposé en équipements ou en machines comme cela est le cas dans les industries manufacturières. Le trafic se représente par des échanges,

dans un même plan ou dans des plans différents, entre des entités de même niveau hiérarchique. Les aspects temporels s'expriment en terme de durée de validité de l'information. Afin d'assurer l'indépendance des fonctions, le modèle impose de ne considérer que des échanges de variables d'états, en excluant toute demande explicite de service. Toutes les fonctions d'une application peuvent recevoir en même temps une certaine information grâce à la possibilité d'utiliser des mécanismes de diffusion.

L'axe vertical représente les niveaux hiérarchiques, comme le fait, par exemple le modèle NBS, mais selon d'autres critères. A un niveau donné, une entité offre des services à l'entité de niveau supérieur en utilisant les services des entités de niveaux inférieurs et vice-versa. Ces échanges, entre deux niveaux adjacents d'une même fonction, se font par l'intermédiaire de requêtes/confirmations si l'initiateur de l'échange est le niveau supérieur, et d'indications/réponses si c'est le niveau inférieur qui est l'initiateur de l'échange [THO 89b], [SIM 92a]. Les aspects temporels s'expriment en terme de temps de réponse. Lors de l'envoi d'une requête, le temps de réponse est garanti si le temps de transmission et si le temps de traitement de la requête le sont également. En général, plus le niveau est faible, plus on se rapproche du niveau zéro des RLI et plus les contraintes de temps de réponse sont sévères.

Le troisième axe représente de façon indépendante les différentes fonctions que l'on retrouve dans une application CIM.

Le modèle trois axes propose une représentation d'architecture fonctionnelle. Il permet de mettre en évidence les différents types de flux d'informations existants entre les entités, ainsi que les différentes contraintes qui leur sont attachées. On retrouve donc un ensemble d'entités et de types de flux possédant chacun leurs propres contraintes.

3. Les réseaux dans les Systèmes Automatisés

La nécessité de faire communiquer un ensemble de fonctions situées sur plusieurs machines différentes impose, suivant le besoin de communication, d'introduire un certain type de RLI (Réseau Local Industriel).
Un RLI permet de faire communiquer les divers composants d'un SI. La nécessité de satisfaire à différents besoins (cohérence, gestion de production, maintenance, simplification du câblage, ...) a conduit à définir différents types de services (téléchargement, interconnexion de systèmes hétérogènes) et différents types de réseaux [THO 86], [THO 89a] que nous présenterons en annexe.

Dans le cadre de la modélisation de la communication dans les SA, on peut classifier les RLI selon trois niveaux distincts, qui répondent chacun à trois

catégories de besoins en communication. Les différents critères de classification des échanges sont la fréquence des informations transmises, les contraintes de temps associées aux échanges d'informations, les fonctions mises en oeuvre, le type et la structure de l'information échangée ainsi que l'environnement dans lequel on se situe. En fonction des différentes caractéristiques des RLI, il est possible d'introduire trois niveaux :

- au niveau zéro, on retrouve les échanges d'informations nécessitant de respecter de fortes contraintes de temps. On se situe dans le domaine des réseaux de terrain. Le besoin de communication est alors très déterministe et une grande partie du trafic se fait de manière périodique. Les échanges d'information sont souvent de faible taille et ils se font essentiellement entre des capteurs, des actionneurs et des automatismes. L'unité de temps généralement employée est la milli seconde ou la dizaine de milli secondes. Citons ici, à titre d'exemple, les réseaux de terrain FIP [UTE 90a], [UTE 90b], [UTE 90c] et InterBus-S [BEN 92].

- au niveau un, l'information est beaucoup plus structurée et les contraintes de temps sont moins strictes que celles du niveau zéro. C'est dans cette partie que l'on retrouve généralement les fonctions de téléchargement, de télélecture de programmes entre des automates, des postes opérateurs et des consoles de programmation (notons que ces fonctions se retrouve également au niveau zéro). L'unité de temps employée se situe entre la milli seconde et la seconde. Citons, par exemple, les réseaux mini-MAP [MAP 88] et PROFIBUS [DIN 90].

- au niveau deux, les informations échangées sont souvent des fichiers et des données volumineuses. On retrouve donc des échanges aléatoires d'informations entre les systèmes de CFAO (Conception et Fabrication Assistées par Ordinateur), de GPAO (Gestion de Production Assistée par Ordinateur) et de CAO (Conception Assistée par Ordinateur). L'unité de temps employée est souvent de plusieurs secondes. Citons, à titre d'exemple, le réseau TOP [ROU 88] qui entre dans cette catégorie.

Un exemple d'illustration des trois niveaux hiérarchiques peut se représenter de la manière suivante:

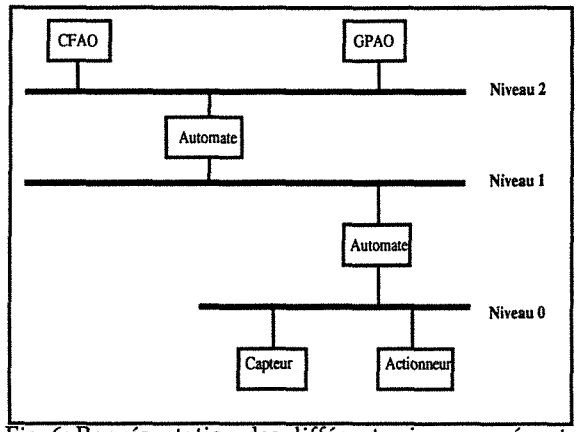


Fig. 6: Représentation des différents niveaux présents dans les réseaux locaux industriels

Les réseaux des niveaux inférieurs sont généralement capables de répondre aux exigences de temps des réseaux de niveaux supérieurs, où les contraintes de temps sont généralement moins fortes [SHI 93].
Par contre, un réseau de niveau supérieur ne peut pas toujours répondre aux exigences temporelles des niveaux inférieurs. Ainsi, par exemple, le réseau MAP ne peut pas satisfaire les contraintes de temps strictes imposées par le niveau zéro [SHI 92].

4. Architecture fonctionnelle, architecture matérielle et architecture opérationnelle

Dans cette partie, nous allons nous intéresser maintenant au passage de l'architecture fonctionnelle à l'architecture opérationnelle.

L'architecture matérielle comprend la liste des différents moyens matériels qui sont mis à disposition, pour mettre en oeuvre les différentes fonctions décrites dans l'architecture fonctionnelle [SIM 92b], [VEG 93].

Des contraintes sont associées aussi bien à l'architecture fonctionnelle (telles que les contraintes de temps) qu'à l'architecture matérielle (telles que les contraintes liées aux constructeurs).

L'architecture opérationnelle est composée de l'architecture du système d'information provenant du résultat de la conception et de la modélisation des solutions retenues lors de la composition de l'architecture fonctionnelle et de l'architecture matérielle. Ainsi, elle prend en compte la liste des machines utilisées, la liste des différents réseaux intervenant pour réaliser un système complet de communication, ainsi que la liste des différentes connexions des machines sur les réseaux.

L'implantation des architectures fonctionnelles en utilisant les architectures matérielles données (processeurs, réseaux, ...) permet d'obtenir des architectures opérationnelles. La figure ci-dessous permet de schématiser les liens existants entre les architectures fonctionnelles, matérielles et opérationnelles:

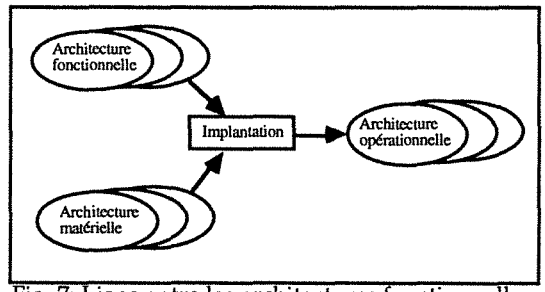


Fig. 7: Liens entre les architectures fonctionnelles, matérielles et opérationnelles

Les contraintes de temps présentées dans les architectures fonctionnelles et matérielles doivent se retrouver dans l'architecture opérationnelle. Ainsi, suivant le type de machine employé, différents mécanismes temporels seront mis en oeuvre dans l'architecture opérationnelle.

Dans l'architecture fonctionnelle, on exprime des contraintes, des propriétés ainsi que les caractéristiques attendues de la solution.

Dans l'architecture matérielle, on retrouve les caractéristiques temporelles provenant du choix de matériels, des réseaux retenus, etc ...

Dans l'architecture opérationnelle, on vérifie que les caractéristiques attendues et que les propriétés soient respectées grâce aux choix et à la répartition des choix faits dans l'architecture matérielle.

Ainsi, dans l'architecture fonctionnelle, les besoins fonctionnels et temporels se traduisent en terme de contraintes de temps.
Les besoins temporels de l'architecture fonctionnelle ainsi que les caractéristiques temporelles provenant de l'architecture matérielle s'expriment en terme de modèles de communication, de services et de protocoles au niveau de l'architecture opérationnelle.

Dans l'architecture opérationnelle, des mécanismes temporels de vérification des contraintes de temps permettent de mettre en oeuvre les modèles de communication, les services et les protocoles retenus, et de vérifier les contraintes de temps exprimées dans l'architecture fonctionnelle.

Le schéma ci-dessous résume les différents concepts temporels introduits dans les architectures fonctionnelles, matérielles et opérationnelles :

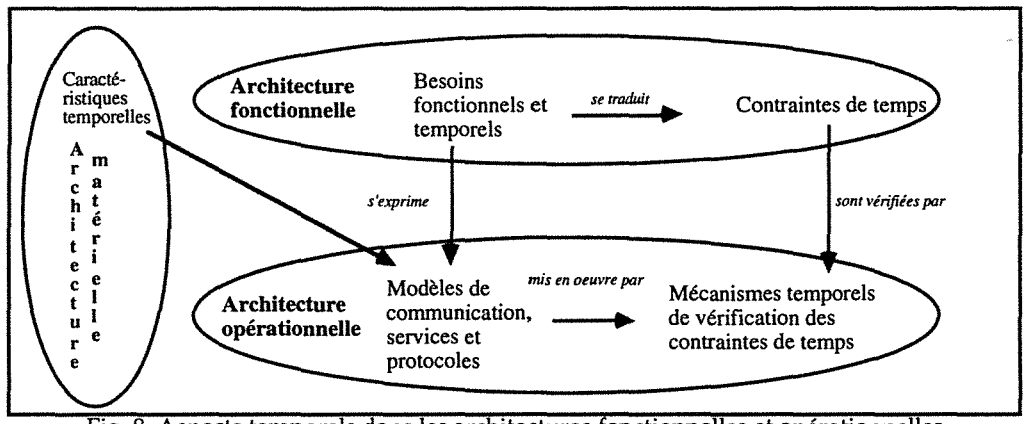


Fig. 8: Aspects temporels dans les architectures fonctionnelles et opérationnelles

Dans le paragraphe suivant, nous allons étudier les modèles de communication client(s)/serveur(s) et producteur(s)/consommateur(s). Notons qu'à partir d'un besoin utilisateur, en fonction des modèles de communication retenus, une transaction donnée va invoquer un ou plusieurs types de services différents qui nécessitent chacun un temps d'exécution qui leur est propre.

Pour un modèle de communication donné, il est possible d'introduire des mécanismes temporels, pouvant se traduire par exemple en terme de machines d'états temporelles, et permettant d'indiquer si un certain besoin temporel, s'exprimant en terme de contraintes de temps, a été respecté ou non durant l'étape de production, de transmission ou de consommation. Dans la suite de notre étude, nous allons nous attacher à définir des mécanismes permettant, lors de l'exécution des processus d'application, de vérifier qu'un événement se produit dans un certain intervalle de temps fixé.

5. Modèles de communication dans les architectures opérationnelles

Nous allons maintenant nous intéresser tout particulièrement, aux aspects temporels dans l'architecture opérationnelle. Pour cela, nous allons tout d'abord présenter les différents types de modèles de communication qui peuvent être utilisés dans les architectures opérationnelles. Ensuite nous étudierons les aspects temporels, en montrant comment la notion de temps intervient dans ces différents modèles de communication.

En fonction du modèle de communication choisi, les services et les qualités de services offerts seront différents. Les modèles de communication que nous allons présenter dans le cadre de notre étude de l'architecture opérationnelle sont les suivants : les modèles de communication producteur(s)/consommateur(s), client/serveur, client/multiserveurs et multiclients/multiserveurs [LOR 94c].

5.1 Modèle Producteur/Consommateur

Dans le modèle de communication producteur/consommateur, le producteur produit une certaine information et met cette information à disposition d'un consommateur qui peut ou non la consommer. Notons que les entités producteur et consommateur se trouvent au même niveau. Une notion importante a prendre en compte dans ce modèle est la notion de validité de l'information disponible.

Le modèle de communication producteur/consommateur suppose une certaine configuration du réseau puisque les producteurs connaissent souvent les besoins des consommateurs, ainsi que la fréquence de ces besoins.

Deux mécanismes différents peuvent être utilisés: le mécanisme de buffer et le mécanisme de file d'attente.

Si c'est le mécanisme de buffer qui est utilisé, alors la production d'une nouvelle valeur écrase l'ancienne valeur qui est considérée comme obsolète. Le producteur a alors toujours, pour une variable donnée, une valeur à émettre vers les consommateurs. Si l'entité de production n'a pas mis à jour son buffer à l'instant t_n, alors l'information émise sur le réseau de communication à l'instant t_n est identique à l'information produite à l'instant t_{n-1}. De même chez le consommateur, si c'est le mécanisme de buffer qui est utilisé, alors il est toujours possible de consommer une valeur. Notons que dans ce cas, deux valeurs consommées de manière consécutive par le processus d'application peuvent être identiques, si aucune nouvelle valeur n'a été reçue depuis la précédente consommation.

Si c'est le mécanisme de file d'attente qui est utilisé, alors lorsque la file d'attente de l'entité de production est vide, alors le producteur émet une trame vide vers les consommateurs. Lorsque l'entité de consommation consomme une valeur, il la supprime de la file d'attente de l'entité de consommation. Donc si la file d'attente de l'entité de consommation est vide, alors la valeur consommée est indéterminée.

5.2 Modèle Producteur/Consommateurs

Dans le modèle de communication producteur/consommateurs, une certaine information produite peut être consommée en même temps et à un instant précis, par plusieurs consommateurs. On s'assure de la bonne réception de l'information dans les délais grâce à des mécanismes permettant de vérifier la

cohérence temporelle (ces mécanismes seront étudiés en détail dans les chapitres suivants). C'est le principe de la diffusion de l'information qui est utilisé dans le modèle de communication producteur/consommateurs: l'échange d'information ne se fait pas en point-à-point mais en multipoints.

La communication dans le modèle producteur/consommateurs peut se schématiser de la manière suivante:

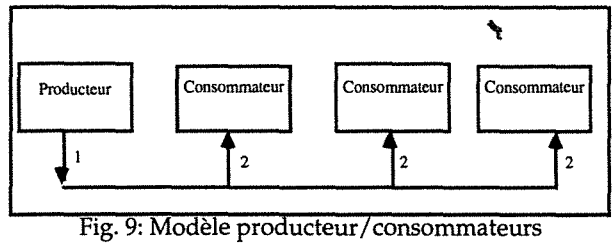


Fig. 9: Modèle producteur/consommateurs

Chaque station peut être à un instant donné, soit producteur, soit consommateur d'une certaine donnée. Plusieurs stations connectées sur le réseau peuvent donc recevoir en même temps la valeur émise par un producteur. Il est possible d'élaborer des mécanismes de cohérence spatiale permettant de s'assurer de l'identité d'une même valeur chez plusieurs consommateurs à un instant donné.

Notons qu'il est possible d'introduire une fonction de distribution permettant à chaque producteur de savoir à quel instant précis, il doit produire une certaine information à destination des différents consommateurs. On obtient alors le modèle PDC (Producteur/Distributeur/Consommateurs) [THO 91b].

Ce modèle peut se représenter de la manière suivante:

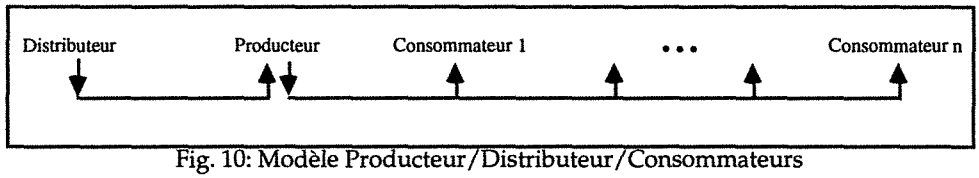


Fig. 10: Modèle Producteur/Distributeur/Consommateurs

Le distributeur permet d'ordonnancer la circulation des données entre le producteur et les consommateurs. Ainsi, si plusieurs consommateurs souhaitent consommer une même information, il ne sera nécessaire de ne faire qu'une seule demande de production.

5.3 Modèle Producteurs/Consommateurs

Lorsqu'un ensemble de règles lient les différents producteurs entre eux, il est nécessaire d'introduire un nouveau modèle de communication (le modèle producteurs/consommateurs) permettant aux producteurs de mettre en oeuvre une opération atomique.

Traitons par exemple le cas où tous les producteurs doivent produire à un même instant précis, une certaine information à destination d'un ensemble de consommateurs. Alors après s'être synchronisés, les producteurs envoient leurs informations, représentatives d'un même instant donné, vers les différents consommateurs qui reçoivent tous l'information en même temps. Ceci nous conduit à introduire le modèle de communication Distributeur/ Producteurs/Consommateurs.

Ce modèle peut se schématiser de la manière suivante:

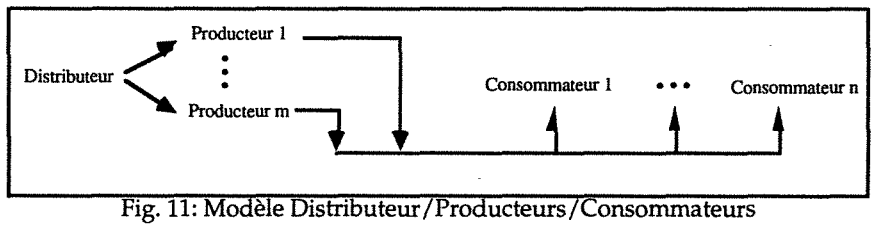


Fig. 11: Modèle Distributeur/Producteurs/Consommateurs

Comme les informations produites par les différents producteurs ne peuvent être transmise que de manière séquentielle, on utilisera pour chaque variable deux buffers: un buffer d'émission et un buffer de réception.
A un instant donné et sur réception d'un ordre de resynchronisation de la part d'un distributeur, tous les producteurs intervenant dans l'élaboration d'une liste de variables font une copie de leur buffer d'émission vers leur buffer de réception. Ils "produisent" ensuite tous en même temps puisque l'information échangée entre les entités productrices et les entités consommatrices concerne des valeurs de variables "figées" qui se trouvent dans les buffers de réception des différents producteurs.
Après la transmission via le médium de toutes les valeurs des variables contenues dans les buffers de réception des entités de production vers les buffers d'émission des entités consommatrices, sur réception d'un ordre de resynchronisation, tous les consommateurs font une copie de leur buffer d'émission vers leur buffer de réception, et ainsi ils peuvent tous consommer en même temps une même liste de variables.

L'échange d'information entre les trois producteurs des variables A, B et C et deux consommateurs de la liste de variables (A, B, C) souhaitant consommer en même temps les valeurs des variables A, B et C produites au même instant, peut se schématiser de la manière suivante:

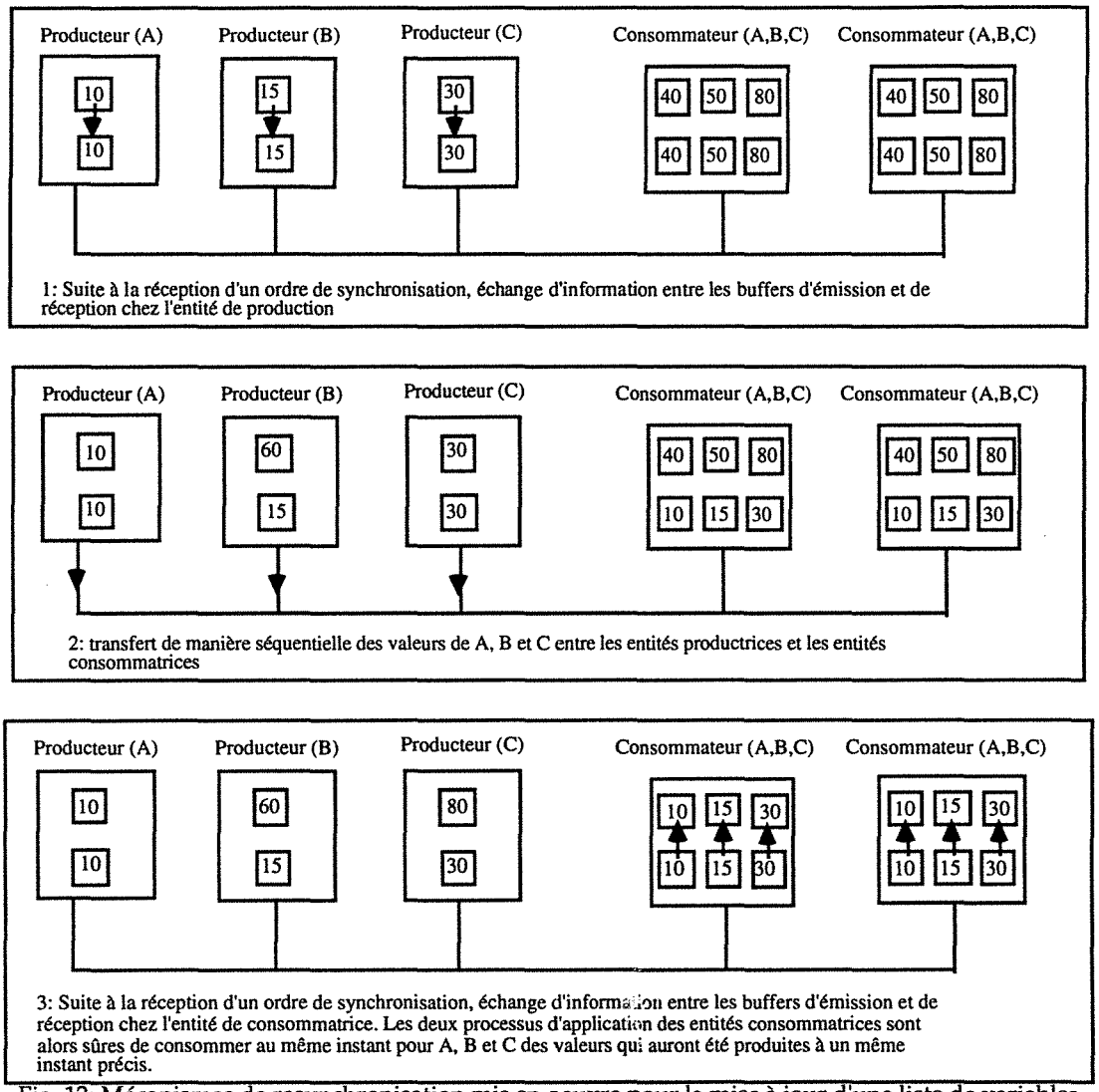


Fig. 12: Mécanismes de resynchronisation mis en oeuvre pour la mise à jour d'une liste de variables

Ainsi, le modèle de communication producteurs/consommateurs permet à une liste de données produites à un même instant t1 par plusieurs producteurs, d'être consommée à un instant précis t2 (avec t2>t1) par plusieurs consommateurs.

Notons que le modèle de communication producteurs/consommateur se déduit directement du modèle de communication producteurs/ consommateurs, il suffit alors de poser dans ce dernier modèle que le nombre de consommateurs est égal à un.

5.4 Le modèle Client/Serveur

Le modèle de communication client/serveur s'applique à une association bipoint où interviennent deux processus d'application, le client et le serveur [ISO 89b]. Le client émet une demande de service à un serveur ; le serveur reçoit alors une indication l'informant d'une demande de service à effectuer. Lorsque le traitement du service par le serveur est terminé, ce dernier envoie la réponse au client et le client reçoit cette réponse sous la forme d'une confirmation.

Dans le modèle de communication client/serveur le temps de traitement, de la requête du client, utilisé par le serveur est souvent indéterminé, puisqu'il dépend fortement de la charge du serveur au moment de la réception de l'indication de demande de service.

Les services mis en oeuvre entre un client et un serveur sont constitués de requêtes de demande, d'indication, de réponse et de confirmation. L'ordonnancement de ces différentes requêtes est représenté sur le schéma ci-dessous:

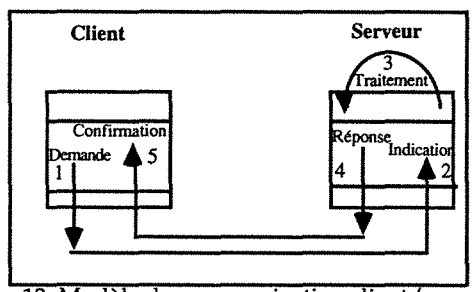


Fig. 13: Modèle de communication client/serveur

Notons que si l'on souhaite que chaque requête véhicule des contraintes temporelles, il est alors nécessaire de construire un nouveau modèle de communication que nous nommerons "client/serveur temporel".

Si à un instant donné, plusieurs clients veulent lire une même valeur chez un même serveur, ils doivent alors envoyer chacun une demande vers le serveur en question. Après le traitement séquentiel de chacune des requêtes, le serveur envoie une réponse à chacun des clients. On ne possède donc aucune garantie concernant l'identité des réponses suite à l'envoi d'une même demande de service effectuée par des clients différents à destination d'un même serveur ; puisque chacune des demandes fait l'objet d'un traitement particulier, à un instant différent, de la part du serveur. Ainsi pour une même variable, les valeurs retournées par le serveur aux différents clients peuvent être différentes les unes des autres.

Comme pour le modèle producteur/consommateur, deux mécanismes différents peuvent être utilisés: le mécanisme de buffer et le mécanisme de file d'attente.
C'est généralement le mécanisme de file d'attente qui est utilisé puisque ce mécanisme permet de prendre en compte les demandes de tous les clients de manière séquentielle. Suite à la réception d'une requête formulée par un client et après traitement de cette requête par le serveur, ce dernier renvoie le résultat au client. Le mécanisme de file d'attente permet à plusieurs requêtes d'être stockées dans une file d'attente avant d'être traitées de manière séquentielle par le serveur.

L'utilisation du mécanisme de buffer dans le modèle client/serveur, impose au serveur de mettre à jour son buffer de manière plus ou moins périodique. Lors d'une demande d'échange d'information de la part du client, la valeur émise est alors directement celle qui se trouve dans le buffer du serveur. Ceci évite donc le temps d'attente et de mise à disposition de l'information par le serveur.
Les mécanismes de buffer imposent au serveur d'avoir un buffer différent pour chaque information produite. De plus, il est nécessaire de configurer le serveur de telle façon, que ce dernier mette à jour les valeurs stockées dans les buffers selon une certaine périodicité et selon un certain ordonnancement, et ceci afin de garantir une certaine fraîcheur de l'information. On pourra utiliser dans ce cas les mécanismes de configuration de gestion de réseau introduits dans [LOR 90], [LOR 91] et [LOR 93a].

Ainsi, deux modèles de communication différents peuvent être élaborés: le modèle client/serveur_buffer et client/serveur_file.
L'intérêt d'utiliser les mécanismes de files d'attente par rapport aux mécanismes de buffer s'explique par le fait que les files d'attente permettent d'être sûr d'avoir la dernière valeur disponible, mais avec un délai correspondant au temps du traitement de la requête par le serveur qui peut être plus ou moins long.
Par contre les mécanismes de buffer permettent d'avoir une réponse immédiate, mais on ne peut pas garantir que la valeur envoyée est bien significative de l'application à l'instant précis de la demande.

Notons que dans le modèle de communication producteurs/consommateurs, la fonction de distribution que nous avons introduite peut être considérée comme un client particulier dont le rôle est de synchroniser des différents serveurs (c'est-à-dire les producteurs).
Lorsque le client envoie un message de demande à un serveur, le client est considéré comme le producteur et le serveur comme le consommateur. De même, lorsque le serveur envoie le résultat du traitement vers le client, alors le serveur est alors considéré comme le producteur et le client comme le consommateur.

Une représentation comparative des modèles de communication producteur/consommateur et client/serveur peut se schématiser de la manière suivante:

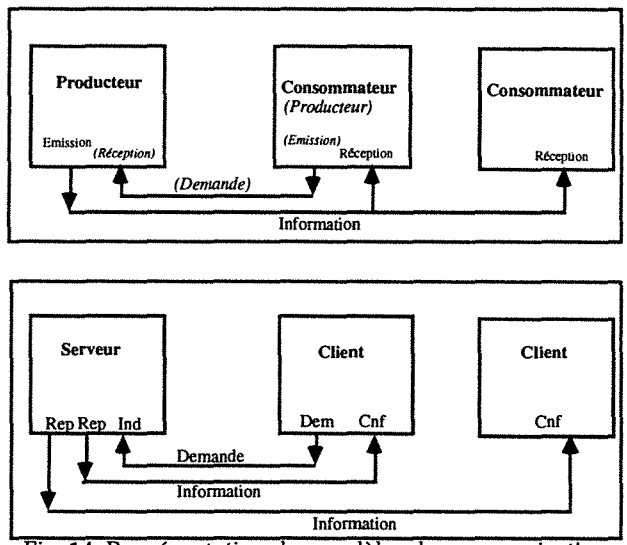


Fig. 14: Représentation des modèles de communication producteur/consommateur et client/serveur

La grande différence entre ces deux modèles de communication repose alors sur le fait que dans le modèle de communication producteur/consommateur, une certaine valeur produite par un producteur donné peut être consommée par différents consommateurs en même temps. Par contre dans le modèle client/serveur, une même réponse est envoyée de manière individuelle par un même serveur vers chacun des clients souhaitant recevoir l'information. Donc dans le modèle de communication client/serveur, on ne peut pas élaborer de statuts temporels indiquant qu'une même valeur a bien été reçue en même temps par plusieurs clients.

5.5 Le modèle Client/Multiserveurs

Dans le modèle de communication client/multiserveurs, plusieurs types de communication sont possibles suivant que l'on combine :

1°) une ou plusieurs requêtes différentes en provenance d'un client avec le fait que
2°) les serveurs peuvent être connus ou inconnus.

On obtient alors les quatre possibilités suivantes:

1.1) une requête en provenance d'un client est satisfaite par plusieurs serveurs connus du client,
1.2) une requête en provenance d'un client est satisfaite par plusieurs serveurs non connus du client [DAK 90],
2.1) plusieurs requêtes en provenance d'un même client sont envoyées à plusieurs serveurs connus du client,
2.2) plusieurs requêtes en provenance d'un même client sont envoyées à plusieurs serveurs non connus du client.

Il est important de noter que pour le client, le fait que les serveurs soient connus ou non, n'influe pas sur les réponses reçues par le client. Nous avons uniquement voulu montrer ici les différents types de communication possibles.

Par la suite, nous allons traiter uniquement le cas où les contraintes sont les plus sévères, c'est-à-dire le cas 2.2, puisque l'on traite plusieurs requêtes et que les serveurs ne sont pas connus. Les autres cas pourront se déduire de cette dernière étude.

Un client envoie ses requêtes à un serveur connu de ce dernier ; ce serveur est alors lui-même client d'autres serveurs qui seront cette fois-ci connus de ce dernier client ; chacun de ces derniers serveurs pourront également être clients d'autres serveurs, etc... Ainsi lorsque les serveurs ne sont pas connus du client initial, il est nécessaire de rajouter au moins une étape supplémentaire d'échange d'information entre un client et un serveur.

Une représentation d'un ensemble de transactions en arborescence peut se schématiser de la manière suivante:

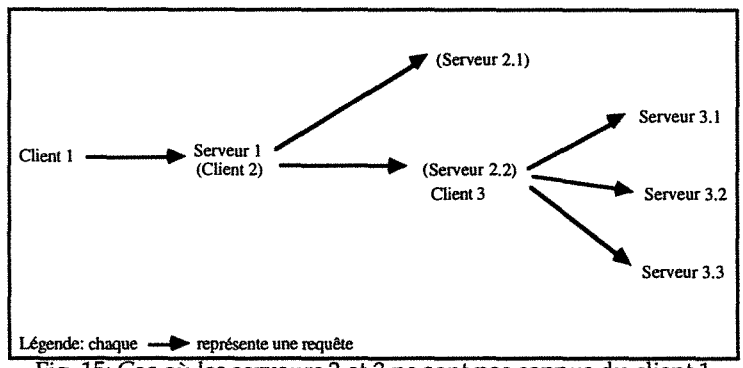


Fig. 15: Cas où les serveurs 2 et 3 ne sont pas connus du client 1

Si on impose que la communication se fasse dans les délais, il faut alors s'assurer que chacun des serveurs a bien répondu dans les délais fixés.

Lorsque l'on numérote de manière chronologique l'ordre des messages échangés entre un client et des serveurs, on constate alors que les serveurs ne répondent pas nécessairement tous en même temps. Ainsi dans le cas où un

client veut communiquer avec trois serveurs connus, il doit envoyer trois requêtes différentes qui n'arriveront pas nécessairement en même temps et qui seront traitées à des instants différents par les différents serveurs, puisque cela dépend de la charge du système de communication et de la charge de chaque serveur à un instant donné [THO 93].

Dans le cas où l'on s'intéresse uniquement à l'ordonnancement des requêtes, il est possible de simplifier notre étude dans le cas où les serveurs sont connus du client ; un exemple d'enchaînement de type client/multiserveurs basé sur le modèle client/serveur peut alors se représenter de la manière suivante:

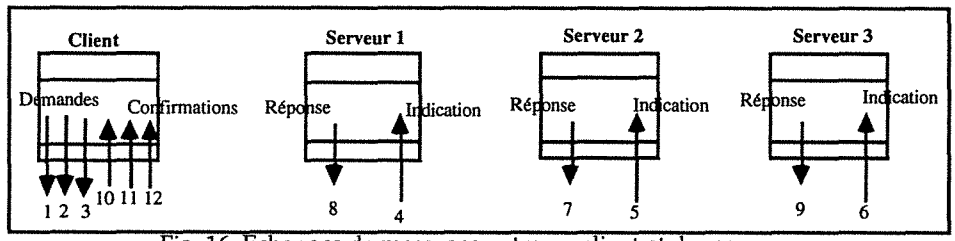


Fig. 16: Echanges de messages entre un client et des serveurs

Dans ce dernier exemple, le client instancie trois demandes de services avec trois serveurs différents. L'ordre des réponses en provenance des trois serveurs (numéroté 7, 8 et 9 sur notre schéma) est connu dans le cas où les serveurs ont la possibilité de répondre immédiatement et si on ne tient pas compte de la charge du réseau de communication. Dans le cas contraire, l'ordre des réponses sera aléatoire puisque chaque serveur aura alors besoin d'un temps qui lui est propre pour traiter une demande de service.

5.6 Le modèle Multiclients/Multiserveurs

Lorsqu'un ensemble de règles lient les différents clients entre eux, il est nécessaire d'introduire un nouveau modèle de communication (le modèle de communication multiclients/multiserveurs) permettant aux différents clients de mettre en oeuvre une opération atomique.

Quatre cas peuvent se représenter suivant que :

1°) les requêtes émises par les différents clients sont identiques ou différentes,
2°) les serveurs sont connus ou inconnus des clients.

On obtient alors les quatre possibilités suivantes:

1.1) les différents clients émettent des requêtes identiques à destination d'un ensemble de serveurs connus des clients,
1.2) les différents clients émettent des requêtes identiques à destination d'un ensemble de serveurs non connus des clients,
2.1) les différents clients émettent des requêtes différentes à destination d'un ensemble de serveurs connus des clients,
2.2) les différents clients émettent des requêtes différentes à destination d'un ensemble de serveurs non connus des clients.

Dans le cas où les différentes requêtes en provenance des clients sont envoyées au même instant, il est alors possible d'élaborer des statuts temporels permettant d'indiquer si les réponses obtenues ont bien été reçues ou non dans un même intervalle de temps.

Le fait que les serveurs ne soient pas connus des clients apporte plus de souplesse au système ; ainsi par exemple, il est possible de changer les adresses des serveurs, en informant uniquement le poste de travail qui a en charge de connaître les différentes adresses des serveurs.

Si on s'intéresse uniquement à l'ordonnancement des requêtes, il est possible de simplifier notre étude dans le cas où les serveurs sont connus des clients ; un exemple de communication entre deux clients et trois serveurs basé sur le modèle client/serveur peut alors se représenter de la manière suivante :

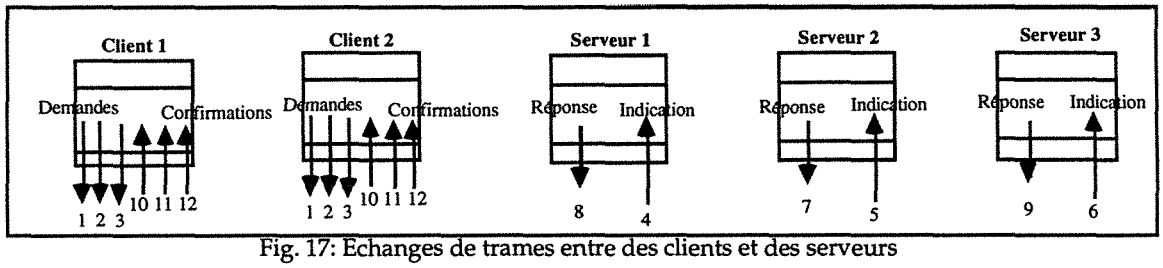


Fig. 17: Echanges de trames entre des clients et des serveurs

Dans ce dernier exemple, deux clients différents font des demandes de services identiques au même instant à trois serveurs. Ces demandes sont numérotées 1, 2 et 3 sur notre schéma. Les réponses des trois serveurs sont diffusées de manière séquentielle vers chacun des deux clients, qui reçoivent alors les mêmes confirmations (numéroté 10, 11 et 12 sur notre schéma). Il est possible de mettre en oeuvre des mécanismes permettant de savoir, lorsque plusieurs clients font des demandes de service identiques à un ensemble de serveurs, si les réponses reçues par ces différents clients sont identiques ou non.

Notons que le modèle de communication multiclients/serveur, se déduit directement du modèle de communication multiclients/multiserveurs, il suffit alors de poser dans ce dernier modèle que le nombre de serveur est égal à un.

6. Conclusion

Dans ce premier chapitre, nous avons développé les aspects de la communication dans les Systèmes Automatisés en s'attachant tout d'abord à la description des différentes architectures fonctionnelles. Les insuffisances des architectures fonctionnelles classiques conduisent à utiliser une architecture fonctionnelle basée sur le modèle trois axes.

Dans une seconde partie, nous nous sommes intéressés au passage de l'architecture fonctionnelle à une architecture opérationnelle en se positionnant tout particulièrement sur l'étude des besoins temporels et des contraintes temporelles. Ensuite, l'étude des modèles de communication dans l'architecture opérationnelle, nous a conduit à présenter les modèles de communication producteur(s)/consommateur(s), ainsi que dans les modèles de communication clients/serveurs, client/multiserveurs, multiclients/multiserveurs.
Le choix d'un modèle de communication se faisant à partir des contraintes de temps énoncées dans l'architecture fonctionnelle, des choix matériels retenus, et en fonction de la qualité de service souhaitée par les utilisateurs.

L'étude des propriétés temporelles des différents modèles de communication introduits dans ce chapitre, sera poursuivi dans les chapitres suivants, et ceci en fonction des résultats obtenus à partir de notre proposition de spécification et de modélisation des contraintes temporelles dans les architectures de communication.

Chapitre 2

Spécification des contraintes temporelles

Chapitre 2

Spécification des contraintes temporelles

1. Introduction

Dans les systèmes temps réel distribués, la cohérence des fonctions dépend non seulement des résultats fournis, mais surtout du respect des contraintes de temps par les fonctions pour produire leurs résultats [THO 90b], [ELL 90], [BER 92].

Un système temps réel réalise un ensemble de fonctions dont les instants de début et/ou de fin d'exécution doivent respecter des contraintes temporelles imposées par le processus physique. "Commencer une opération après un instant précis et la finir avant une échéance fixée" est la forme la plus utilisée pour spécifier des contraintes sur l'exécution des opérations en temps réel. Les systèmes qui doivent impérativement respecter des contraintes temporelles strictes pour éviter des catastrophes sont appelés systèmes temps critique.

Pour des raisons d'ouverture des systèmes et des applications, il est recommandé d'utiliser des normes faisant référence au modèle de communication OSI. Mais c'est un modèle général n'intégrant pas le temps sous une forme permettant de s'assurer de la validité temporelle des données [SHE 90]. Il faut donc définir de nouveaux mécanismes à intégrer dans les services et protocoles qui prennent en compte les besoins de la communication pour les systèmes temps critique. Il est important de s'intéresser aux travaux existants concernant la spécification des besoins temporels [NAI 92], le développement de protocoles [RAM 89] permettant de gérer les contraintes de temps [MAR 88] pour répondre aux besoins temporels dans les systèmes temps réel [SCH 92]. Nous proposons d'introduire des mécanismes temporels permettant d'indiquer si les contraintes de temps ont été respectées dans un système temps critique s'appuyant sur le modèle de communication OSI ou sur un modèle en couches.

Les mécanismes temporels doivent permettre d'indiquer si les contraintes temporelles sont respectées ou non, et quand elles ne le sont pas, ces mêmes mécanismes doivent permettre d'indiquer les causes de leur non respect.
Ainsi, par exemple, on devra être capable de savoir, pourquoi dans une installation industrielle, lors du dépassement d'un seuil critique de température détecté par un capteur à l'instant t1, un certain actionneur n'a pas enclenché la fermeture d'une vanne à l'instant $t1+\Delta t$ comme cela était prévu dans le cahier des charges. Est-ce que cela est dû au fait que le capteur n'a pas transmis dans les délais l'information de dépassement de seuil, est-ce que cela est dû à l'encombrement du réseau ou est-ce que cela est dû à l'actionneur qui n'a pas pris en compte l'information reçue dans les délais ?

2. Notions de contraintes temporelles

Dans le rapport du groupe TCCA (Time Critical Communication Architecture) de l'ISO, la communication temps critique est définie de la manière suivante [ISO 93]: "*When one or more application process which sends a message requires it to be received (or received and acted upon, or received and acted upon and confirmed) within a certain bounded time period "window" after its send request to the system*".
Pour parvenir à respecter les contraintes temporelles, dans les systèmes temps critique, il faut s'assurer à la fois que de l'ordonnancement des tâches et des messages respectent les contraintes de temps. Notons que l'ordonnancement des tâches ainsi que l'ordonnancement des messages sont de deux approches complémentaires.

L'ordonnancement de processus temps critique consiste à élaborer la stratégie d'allocation du (ou des) processeur(s) aux processus afin de garantir le respect des contraintes de temps [XU 91]. A ce sujet beaucoup de travaux ont été développés dans la littérature pour proposer des algorithmes d'ordonnancement dans un contexte temps critique [ALA 92], [CAR 93b], [GOT 93]. Des études sont menées sur l'ordonnancement des tâches apériodiques [SPR 89], sur les systèmes de contrôle [CHE 90], ainsi que sur les mécanismes de synchronisation dans les systèmes temps réel [RAJ 91], [NOR 93].

La deuxième approche consiste à s'intéresser à la communication des messages donc aux délais de transfert des messages entre les processus temps critique dans un système réparti [RAY 91]. Il est alors nécessaire d'étudier en détail les aspects temporels dans la communication et d'introduire des modèles de communication du type producteur/consommateur, client/serveur, ... ainsi que des mécanismes temporels permettant de gérer le temps [JUA 92]. Il existe de nombreuses études concernant l'expression du concept de temps [MOT 92], [KOP 92], [SAH 92], des contraintes de temps [DAS 85], de la gestion des

contraintes de temps [RAJ 92], [SAH 88], ainsi que des propositions de protocoles de communication intégrant le temps [SON 91], [SIM 90]. Il est important de souligner l'importance des applications de communication mises en oeuvre dans les systèmes distribués [PER 85], [LEL 91], [LEL 86]. Historiquement l'ordonnancement des tâches a été la première approche à être développée, donc cela explique le fait que les algorithmes d'ordonnancement des messages étudiés actuellement dans la communication temps critique s'inspirent largement des algorithmes d'ordonnancement des tâches.

Les délais de communication dans les réseaux sont en général non déterministes, et par conséquent les données arrivant chez un consommateur peuvent être périmées puisque, dans un contexte temps critique, les données ont une durée de vie (ou durée de validité) déterminée par la nature de l'application [KOP 90]. La durée de vie d'une donnée chez le producteur (resp. le consommateur) représente l'intervalle de temps durant lequel la valeur produite (resp. consommée) reste valide et a donc un sens pour l'application.

Pour illustrer la notion de durée de vie, prenons un exemple de processus périodiques. Soit un capteur qui produit une mesure toutes les 20 ms. Les mesures effectuées sont émises vers deux stations consommatrices, un régulateur et un superviseur. Le régulateur consomme les données du capteur avec une période de 10 ms et le superviseur consomme les données du capteur avec une période de 1 s. La durée de vie d'une mesure est fixée à 20 ms pour le régulateur et à 1 s pour le superviseur. On peut alors rapidement constater que le régulateur et le superviseur ne consomment pas nécessairement les mêmes valeurs. Le régulateur doit être paramétré pour consommer les valeurs produites par le capteur uniquement toutes les 20 ms parce que le capteur ne produit une nouvelle mesure que toutes les 20 ms. Mais ceci ne gêne pas le régulateur puisque la valeur consommée par ce dernier reste valide pendant 20 ms, ce qui correspond à la période de production.

Ainsi les données produites sont valides et elles n'ont un sens que pendant un certain intervalle de temps après leur production [LIE 92]. Si on suppose les productions et les consommations périodiques, alors la validité des données consommées dépend essentiellement de trois facteurs : la période de production, la période de consommation et la durée de vie des données.

Nous avons choisi de représenter les délais intervenant durant l'étape de communication en terme de contraintes temporelles. Nous nous intéresserons alors aux contraintes de date au plus tôt et aux contraintes de date au plus tard. Les contraintes temporelles peuvent être fixées par l'application soit de façon dynamique, soit de façon statique lors de l'étape de configuration du réseau. Nous introduirons ensuite des mécanismes permettant de savoir, si au cours de la communication, une certaine donnée a bien été produite, transmise et

consommée dans les délais fixés et dans le cas contraire, ils permettrons de localiser les causes du non respect des contraintes de temps.

Dans un premier temps, nous allons étudier les mécanismes temporels intervenant dans le cadre de la communication entre un producteur et un consommateur. Nous montrerons ensuite comment les mécanismes temporels introduits peuvent être réutilisés pour d'autres types de communication, tels que pour la communication basée sur le modèle de communication client/serveur.

3. Spécification des contraintes temporelles

Il existe différentes possibilités pour spécifier des contraintes de temps, citons par exemple les logiques temporelles [LAM 83], [AUD 91], [OST 91] et les langages spécifiques [KIR 91], [ZAK 84].

Rappelons que la logique temporelle comprend :

- la logique modale qui permet d'exprimer des contraintes de manière formelle par l'intermédiaire d'un certain nombre de symboles,
- les logiques réifiées comprenant les algèbres d'instants et d'intervalles [RAZ 89], [NIC 91], [LAD 86], [SCH 83].

Une spécification à l'aide de la logique modale se représente par un ensemble de formules. La manipulation du temps se fait alors par l'intermédiaire de règles d'inférence. Le temps est alors considéré comme une modalité que l'on associe à un domaine d'interprétation d'une formule [WIL 92]. Cette logique permet, par l'intermédiaire d'un ensemble d'opérateurs, de lier et de décrire différents états ; mais elle ne permet pas de manipuler des notions de date ou de durée [GOD 93], [BRO 89].

L'algèbre d'instants associe des faits et des propriétés à des instants. Un fait est défini par l'ensemble des instants pendant lequel il est vrai [DER 82]. Les seules relations introduites dans l'algèbre d'instants sont les relations de précédence (avant, après, égal) qui permettent, par exemple, de savoir si un fait est vrai dans un intervalle de temps donné.

3.1 L'algèbre d'intervalles

L'algèbre d'intervalles associe des propriétés, des événements et des processus à un intervalle [ALL 83].
Les différentes possibilités de combinaison entre deux intervalles temporels sont décrites dans [ALL 84] et sont représentées dans le schéma ci-dessous:

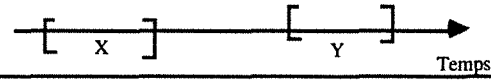
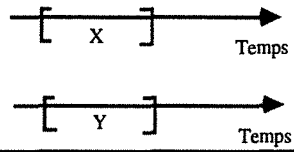
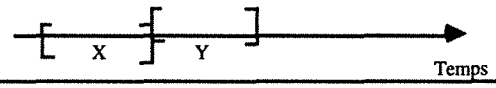
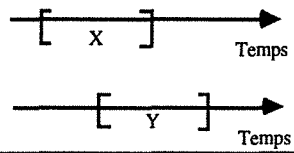
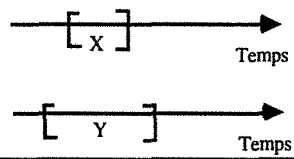
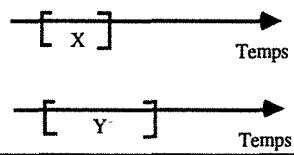
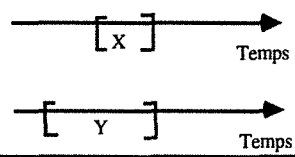
Opérateurs	Exemples
X before Y	
X equal Y	
X meets Y	
X overlaps Y	
X during Y	
X starts Y	
X finishes Y	

Fig. 1: Relations définies par l'algèbre d'intervalles

Les différents types d'opérateurs introduits par l'algèbre d'intervalles vont permettre de prendre en compte les contraintes de temps que peut énoncer l'utilisateur comme par exemple, le fait qu'une variable doit être produite dans un intervalle de production et que la durée de vie de cette variable (représentée également par un intervalle) est contenue ou chevauche l'intervalle de production. Comme les contraintes temporelles de date au plus tôt et de date au plus tard peuvent se représenter en terme d'intervalles temporels, la logique d'intervalles permet alors de représenter les différents liens possibles entre ces

contraintes de temps. Nous retiendrons donc, par la suite, la logique des intervalles et nous désignerons par le terme de **fenêtre temporelle** un intervalle de temps.

3.2 Remarques

Notons l'existence de travaux concernant le développement de méthodes de représentation mixtes intégrant à la fois l'algèbre d'instants et l'algèbre d'intervalles [MEI 91]. De plus, des travaux concernant l'intégration des informations quantitatives et qualitatives du temps dans le cadre de l'intelligence artificielle sont en cours [TOL 92], [HAT 91].

4. Fenêtres temporelles dans la communication

L'intervalle de temps à respecter pour communiquer entre des entités productrices et des entités consommatrices peut être découpé en plusieurs sous-intervalles de temps pour chacune des différentes couches du modèle OSI. Si on affecte un délai à chaque sous-intervalle, il est alors possible de s'assurer du respect des contraintes temporelles et de pouvoir localiser de manière précise, la cause d'un éventuel non respect de ces contraintes temporelles. Ainsi, on est capable de savoir si c'est la production, la communication entre les couches du modèle OSI (aussi bien chez l'entité productrice que chez l'entité consommatrice de l'information) ou la consommation qui ne s'est pas faite dans les délais initialement impartis.

Pour les communications périodiques, il est souvent facile de déterminer les dates au plus tôt et au plus tard à partir de la période et de la durée de communication lorsque cette dernière est connue. Par contre, pour les communications apériodiques (ou aléatoires), on ne connaît pas a priori les instants de début de communication, mais une fois que le début d'une communication apériodique a eu lieu, si on connaît la durée de la fenêtre temporelle, il est possible de connaître la date au plus tard [MAL 92].

Dans certaines applications, il peut être nécessaire de manipuler uniquement des contraintes de dates au plus tôt, mais pas nécessairement de dates au plus tard. Par exemple, dans le cas d'une usine fonctionnant uniquement durant la journée et dans laquelle un entretient a lieu tous les matin de 8 heures à midi ; les contraintes temporelles de date au plus tôt imposent alors à l'installation industrielle de ne jamais fonctionner avant midi sous peine de mettre en péril l'installation et des vies humaine.

Ainsi, on peut avoir des fenêtres temporelles fermées (c'est-à-dire, avec dates au plus tôt et au plus tard) et des fenêtres temporelles ouvertes (c'est-à-dire, sans date au plus tôt ou sans date au plus tard).
Des notations pour les fenêtres temporelles seront introduites dans le paragraphe suivant.

Pour mieux maîtriser les contraintes temporelles, nous décomposons l'échange entre les entités en plusieurs étapes séquentielles qui sont respectivement :

- production de l'information,
- traversée des différentes couches du modèle OSI chez l'entité productrice,
- émission, propagation sur le médium et réception,
- traversée des différentes couches du modèle OSI chez l'entité consommatrice,
- consommation de l'information.

La notion de statut temporel va être utilisée pour indiquer si les contraintes de temps ont été respectées. Ainsi, nous associons à chaque étape de la communication, des statuts temporels permettant de savoir si les contraintes temporelles ont été satisfaites ou non, ainsi que les causes de l'éventuel non respect des contraintes de temps. Ces statuts, associés à une étape de la communication, sont élaborés par une entité de contrôle du respect des contraintes temporelles.

Le fait d'élaborer les statuts temporels au plus haut et au plus bas niveau de la communication (c'est-à-dire au niveau de la couche application et au niveau du MAC) permet de savoir si la production, la traversée des différentes couches du modèle OSI, la transmission sur le médium et la consommation se sont déroulées dans les délais fixés.

Nous utiliserons le terme de fenêtre temporelle de communication pour représenter les différents intervalles de temps intervenant dans la communication entre deux entités. Une fenêtre temporelle de communication peut être découpée en plusieurs sous-fenêtres temporelles de communication, comme par exemple, une première sous-fenêtre de traversée des couches 7 à 3 et une seconde sous-fenêtre de traversée des couches 3 au MAC. Des délais différents sont nécessaires pour traverser les différentes couches, ainsi en ce qui concerne les couches 3, 4 et 5 du modèle OSI, ces délais proviennent essentiellement du temps utilisé pour l'obtention des ressources. Notons qu'au niveau de la communication, il est possible d'avoir une fenêtre temporelle de communication pour chacune des couches du modèle OSI.

Nous utiliserons par la suite, le terme de fenêtre temporelle de production (resp. de consommation) pour indiquer si le producteur (resp. le consommateur) a bien produit (resp. consommé) dans les délais.
Le terme de fenêtre temporelle de transmission sera utilisé pour désigner une fenêtre temporelle de communication allant du MAC de l'entité de production au MAC de l'entité de consommation.
Par la suite, chez l'entité de production (resp. de consommation), nous n'utiliserons qu'une seule fenêtre temporelle de communication pour la traversée des couches allant de la couche application au MAC (resp. du MAC à la couche application). Ainsi, le terme de fenêtre temporelle d'émission sera utilisé pour désigner la fenêtre temporelle regroupant les intervalles de temps nécessaires à la traversée des différentes couches du modèle OSI chez l'entité de production. De même, le terme de fenêtre temporelle de réception sera utilisé pour désigner la fenêtre temporelle regroupant les intervalles de temps nécessaires à la traversée des différentes couches du modèle OSI chez l'entité de consommation.

La figure suivante représente les différentes étapes de la communication à travers un découpage utilisant des fenêtres temporelles de production, d'émission, de transmission, de réception et de consommation:

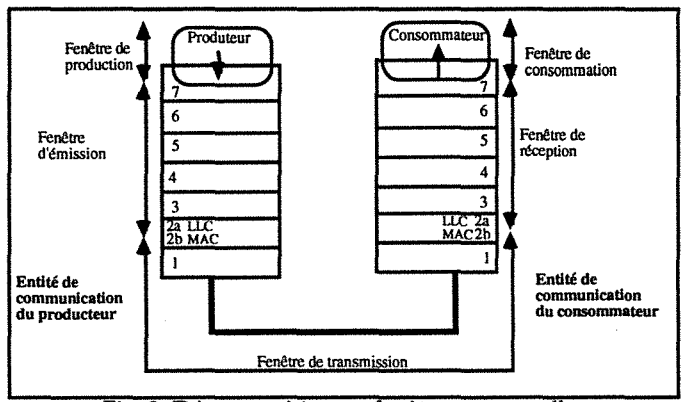


Fig. 2: Décomposition en fenêtres temporelles des différentes phases de la communication

En fonction du découpage des intervalles de temps intervenant dans la communication, on peut définir de manière précise les fonctionnalités des statuts associés aux différentes fenêtres temporelles.

Les statuts de production indiquent si une certaine donnée a été mise à disposition par le producteur dans les délais. Ces statuts seront élaborés chez le producteur par une machine d'états, que nous introduirons dans le chapitre suivant, qui va permettre de savoir si les contraintes temporelles de production ont été respectées. Ces contraintes temporelles sont fixées et liées, pour chaque variable, par configuration de manière statique ou dynamique. Le non respect

des contraintes de production peut être dû à l'une des causes suivantes : le producteur n'a pas reçu l'ordre de produire, il ne peut plus produire dans les délais ou il ne peut plus produire du tout (par exemple, un capteur qui tombe en panne ne peut plus envoyer de mesures).

Les statuts d'émission chez le producteur et les statuts de réception chez le consommateur permettent de savoir si la traversée des couches du modèle de communication OSI et si la transmission sur le médium d'une valeur s'est faite ou non dans les délais. Les statuts d'émission et de réception seront élaborés par une machine d'états, que nous introduirons dans le chapitre suivant, qui va permettre de savoir si les contraintes temporelles associées à la traversée des couches du modèle OSI ont été respectées. Les statuts temporels d'émission vont permettre, par exemple, de savoir si les données ont été émises sur le réseau dans les délais. Les statuts temporels de réception vont permettre de savoir, par exemple, si les données sont arrivées ou non dans les délais, donc si le débit du réseau est suffisant ou non pour répondre aux contraintes de transmission imposées par l'application, si le réseau est surchargé et ne peut plus répondre aux contraintes temporelles, si le réseau est hors service, etc...

Les statuts de consommation indiquent si une certaine donnée a été mise à disposition du consommateur dans les délais. Ces statuts seront élaborés chez le consommateur par une machine d'états, que nous introduirons dans le chapitre suivant, qui va permettre de savoir si les contraintes temporelles de consommation ont été respectées. Ces statuts vont permettre de savoir, par exemple, si le consommateur n'a pas reçu l'ordre de consommer, s'il n'a reçu aucune valeur à consommer (par exemple un actionneur qui tombe en panne ne peut plus consommer les valeurs reçues).

Les différents statuts temporels doivent donc permettre de connaître les causes du non respect des contraintes de temps.

Sous forme algorithmique, il est possible d'exprimer les liens entre les délais de communication et les différents statuts temporels de la manière suivante:

si les contraintes de production des données ont été respectées ***alors***
le statut de production = vrai
sinon
le statut de production indique la cause du non respect des contraintes de temps

si chez l'entité productrice, les contraintes de traitement des données par les différentes couches se trouvant entre la couche application et le MAC ont été respectées ***alors***
le statut d'émission = vrai

sinon

le statut d'émission indique la cause du non respect des contraintes de temps

si *chez l'entité consommatrice, les données ont été reçues dans les délais et si les contraintes de traitement des données par les différentes couches se trouvant entre le MAC et la couche application ont été respectées* ***alors***

le statut de réception = vrai

sinon

le statut de réception indique la cause du non respect des contraintes de temps

si *les contraintes de consommation des données ont été respectées* ***alors***

le statut de consommation = vrai

sinon

le statut de consommation indique la cause du non respect des contraintes de temps.

Notons que la fenêtre temporelle de transmission correspond à l'intervalle de temps délimité d'une part, par la fin de l'étape d'émission et d'autre part, par le début de l'étape de réception. Par conséquent, les contraintes temporelles de transmission peuvent se déduire en fonction du respect des contraintes temporelles de réception de date au plus tôt par le MAC de l'entité consommatrice.

Les statuts temporels de production et d'émission (notés respectivement **SP** et **SE**), concernant le respect, ou non, des contraintes de temps chez l'entité de production sont transmis en même temps que les données vers les entités de consommation. De même chez l'entité de consommation, les statuts temporels de réception et de consommation (notés respectivement **SR** et **SC**), indiquant si les contraintes de temps chez l'entité de consommation ont été ou non respectées, se rajoutent aux données et aux statuts reçus de la part de l'entité de production. L'élaboration des différents statuts temporels au cours de la communication peut se représenter de la manière suivante:

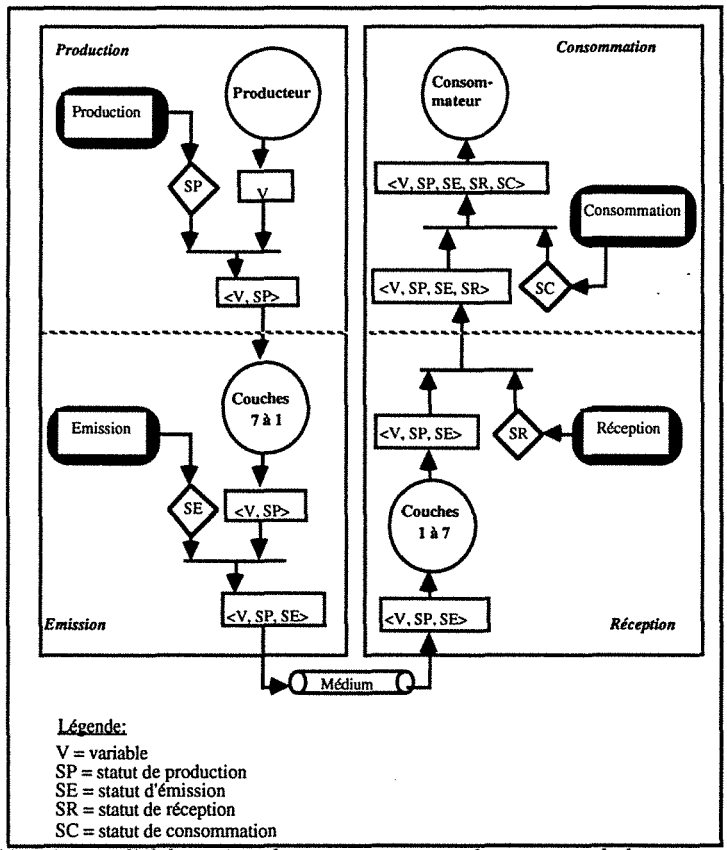


Fig. 3: Mécanismes d'élaboration des statuts temporels au cours de la communication

Les différents statuts temporels de production, d'émission, de réception et de consommation peuvent soit être rajoutés comme paramètres aux services déjà existant, soit provenir de nouveaux services que l'on rajoute aux services existant.

Nous allons maintenant nous intéresser à la définition formelle des concepts de fenêtre temporelle et de durée de vie des données produites et consommées.

4.1 Notations pour les fenêtres temporelles

Le mécanisme de fenêtre temporelle permet de prendre en compte les contraintes temporelles de date au plus tôt et de date au plus tard, ainsi que les contraintes temporelles exprimées sous la forme d'une durée maximale à respecter pour communiquer entre des entités [MAM 94].

Hypothèse: *Par la suite, nous supposerons toujours que les horloges locales des entités productrices et des entités consommatrices sont synchronisées, c'est-à-dire que la différence de temps entre toutes les entités productrices et consommatrices est négligeable devant la nature de l'application. Dans le cas contraire, il n'est pas possible*

de manipuler des contraintes de temps car, par exemple, le fait que l'instant de production d'une donnée est supérieur à l'instant de consommation de cette même donnée n'est absolument pas acceptable : on ne peut pas consommer quelque chose qui n'a pas encore été produit. Ainsi, les débuts et fins des fenêtres temporelles seront fixés en utilisant des horloges physiques synchronisées [LAM 78], [KOP 87], [HE 93].

Une fenêtre temporelle [ISO 93], notée **FT** par la suite, possède trois paramètres qui sont: l'instant de début, l'instant de fin et la durée. Ces trois paramètres peuvent être préétablis ou non définis a priori.

Notons $\uparrow$ l'instant de début, $\downarrow$ l'instant de fin et Δ la longueur (ou la durée) de la fenêtre temporelle.

Soit x une fenêtre temporelle, si on connaît tous les paramètres, on peut alors vérifier que $\uparrow x + \Delta x = \downarrow x$.

4.2 Fenêtres temporelles d'activité

4.2.1 Notations

Dans le cadre de la communication entre différentes entités, nous utiliserons le terme de fenêtre temporelle d'activité, noté **FA**.

Les quatre types de fenêtres temporelles d'activité que nous utiliserons sont les suivants:

- les fenêtres temporelles d'activité de production que nous notons **prod**,
- les fenêtres temporelles d'activité d'émission que nous notons **emis**,
- les fenêtres temporelles d'activité de réception que nous notons **rec**,
- les fenêtres temporelles d'activité de consommation que nous notons **cons**.

Nous utiliserons les paramètres **i** et **v** pour représenter la $i^{ème}$ fenêtre temporelle d'activité invoquée pour la transmission d'une variable v.

Pour chaque nouvelle opération de production, d'émission, de réception ou de consommation d'une variable v donnée, on utilise une nouvelle fenêtre temporelle d'activité en incrémentant l'indice i de la FA ; on passe alors d'une FA avec comme paramètre (v, i) à une nouvelle FA avec comme paramètre (v, i+1).

Une fenêtre temporelle d'activité se décrit alors de la manière suivante:

x y (v, i) avec x = (↑, ↓ ou Δ) et y = (prod, emis, rec ou cons).

Nous allons maintenant poser un ensemble de règles qui devront toujours être prises en compte afin de permettre que la construction ainsi que l'enchaînement des diverses FA se fassent de manière cohérente. Il devra donc être possible de vérifier, par exemple, que pour la $i^{ème}$ fenêtre temporelle d'activité concernant une variable v donnée, le début de la fenêtre temporelle d'activité de consommation ne se situe pas avant le début de la fenêtre temporelle de production.

4.2.2 Enchaînement des fenêtres temporelles d'activité

L'utilisateur fixe généralement les caractéristiques de la FA de production et de la FA de consommation, ou alors ces dernières peuvent être déduites respectivement de l'ordonnancement des sites producteur et consommateur.
Par contre, les caractéristiques concernant l'implémentation des FA d'émission et de réception sont dérivées des caractéristiques des FA de production et de consommation.
Dans ce paragraphe, nous allons donc introduire un ensemble de règles permettant d'enchaîner les différentes FA.

Le fait que le début d'une FA soit toujours antérieur à la fin de cette même FA (de production, d'émission, de réception ou de consommation) se traduit par la règle suivante:

R1: $\uparrow y(v, i) < \downarrow y(v, i)$ avec y = (prod, emis, rec ou cons)

Par la suite nous poserons que :

- l'instant de début (resp. de fin) de la FA de production précède l'instant de début (resp. de fin) de la FA d'émission,
- l'instant de début (resp. de fin) de la FA d'émission précède l'instant de début (resp. de fin) de la FA de réception,
- l'instant de début (resp. de fin) de la FA de réception précède l'instant de début (resp. de fin) de la FA de consommation,

On obtient alors les règles R2 et R3 suivantes :

R2: $\uparrow prod(v, i) \leq \uparrow emis(v, i) \leq \uparrow rec(v, i) \leq \uparrow cons(v, i)$
R3: $\downarrow prod(v, i) \leq \downarrow emis(v, i) \leq \downarrow rec(v, i) \leq \downarrow cons(v, i)$

Dans le cas général et lorsqu'il n'y a pas de problème, les différentes actions entre le producteur et les consommateurs se décrivent de la manière suivante :

- élaboration d'un ordre local ou distant "début-production", noté $\uparrow prod(v, i)$, pour indiquer l'instant à partir duquel le producteur peut commencer à produire des données. Dès qu'une donnée est produite en respectant les contraintes de temps fixées, alors le statut de production prend la valeur Vrai ;

- élaboration d'un ordre local ou distant "début-émission", noté $\uparrow emis(v, i)$, pour indiquer l'instant à partir duquel on peut commencer à transmettre les données à travers les différentes couches du modèle OSI chez l'entité de production ainsi que sur le médium de communication. Dès qu'une donnée est transmise en respectant les contraintes de temps fixées, alors le statut d'émission prend la valeur Vrai ;

- élaboration d'un ordre local "début-réception", noté $\uparrow rec(v, i)$, pour indiquer l'instant à partir duquel on commence à recevoir des données. Dès qu'une donnée est reçue en respectant les contraintes de temps fixées, alors le statut de réception prend la valeur Vrai ;

- élaboration d'un ordre local ou distant "début-consommation", noté $\uparrow cons(v, i)$, pour indiquer l'instant à partir duquel on peut commencer à consommer une donnée. Dès que la donnée est consommée en respectant les contraintes de temps fixées, alors le statut de consommation prend la valeur Vrai.

Un échange périodique de variables signifie que l'entité de production émet de manière périodique des variables qui sont consommées par des entités consommatrices ; donc les FA sont générées de manière statique. Par contre dans le cas d'un échange apériodique, les FA sont générées de manière dynamique.

Si on manipule uniquement des contraintes de date au plus tôt, il n'est pas nécessaire d'enchaîner au mieux les diverses FA, puisqu'il suffit uniquement de commencer à temps. Par contre, dans le cas où l'on a des contraintes de date au plus tard, il va falloir s'assurer du bon respect des contraintes de temps de

toutes les FA et ceci quelque soit l'instant à partir duquel on commence à produire.

Le séquencement des différentes FA peut se faire en utilisant soit des FA fixes, soit des FA glissantes. Les FA fixes permettent le respect des contraintes de dates au plus tôt et au plus tard chez les entités productrices et consommatrices. Par contre, les FA glissantes permettent uniquement le respect des dates au plus tard, mais en offrant la possibilité d'enchaîner au mieux les diverses fenêtres temporelles chez les entités productrices et consommatrices.

Ces dernières peuvent se définir de la manière suivante:

- FA fixes : le début et la fin des intervalles de temps sont fixés une fois pour toutes. Ainsi, chez l'entité de production, que l'on produise plus ou moins tôt dans l'intervalle de temps alloué à la production n'influe pas sur la date de début d'émission de la variable produite. De même, chez l'entité de consommation, le fait qu'une variable soit reçue plus ou moins tôt dans la FA de réception, n'a aucun effet sur l'instant de début de consommation de la variable reçue. Dans le cas de l'enchaînement des FA fixes, l'instant de fin de production d'une donnée précède l'instant de début d'émission de cette même donnée. De même l'instant de fin de réception d'une donnée précède l'instant de début de consommation.

- FA glissantes : dans le cas des FA glissantes, on ne fixe pas de contraintes de temps concernant le début des différents intervalles de temps intervenant dans la communication. Ainsi, chez l'entité de production, dès qu'une donnée est produite en respectant les contraintes de temps, on offre la possibilité de l'émettre en ouvrant une FA dynamique d'émission ; l'ordre de "début-émission" est donc lié à la production d'une donnée. De même, chez l'entité de consommation, on ne fixe pas de contraintes de temps concernant la génération de l'ordre de "début-consommation". Ainsi, chez l'entité de consommation, dès qu'une donnée est reçue en respectant les contraintes de temps, le consommateur peut la consommer puisque l'ordre de "début-consommation" est généré immédiatement après la réception d'une donnée. On constate donc que les FA d'émission et de consommation sont définies de manière dynamique.

Les deux cas d'enchaînement de FA fixes et les FA glissantes peuvent alors se représenter de la manière suivante:

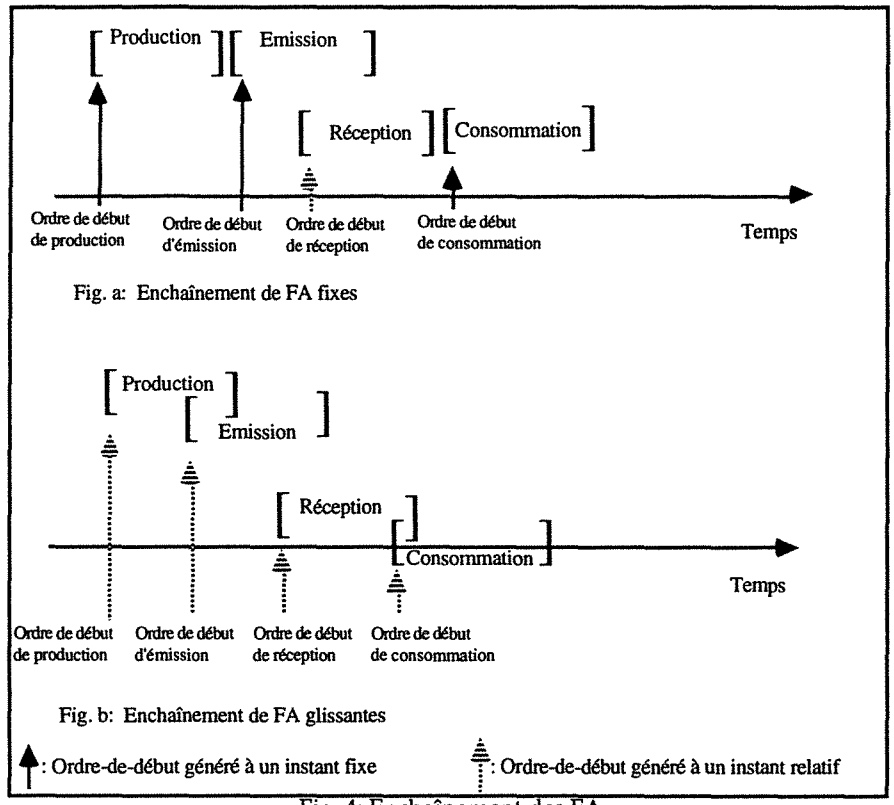


Fig. 4: Enchaînement des FA

Notons que dans le cas des fenêtres temporelles glissantes, le début de la FA d'émission, noté ↑émis(v, i), correspond exactement à l'instant de production de la variable v. De même, le début de la FA de consommation, noté ↑cons(v, i), correspond exactement à l'instant de réception de la variable v.

Aussi bien pour les FA fixes que pour les FA glissantes, la FA de réception chevauche la FA d'émission puisque dès que l'entité de production émet sur le médium de communication, l'entité de consommation doit pouvoir être capable de recevoir les données. La FA de réception est donc définie dynamiquement lors de la réception de données chez l'entité de consommation.

Notons qu'il est possible d'avoir des FA fixes chez l'entité productrice et des FA glissantes chez l'entité consommatrice et vice-versa. Ceci peut s'avérer utile dans le cas où, par exemple, l'entité de production impose le respect de contraintes temporelles strictes concernant le début de l'étape d'émission alors que l'entité de consommation n'impose pas de contraintes temporelles concernant l'étape de début de consommation.

Si on utilise la logique d'intervalles introduite précédemment [ALL 83], l'enchaînement des FA fixes représenté dans la figure 4.a peut se décrire de la manière suivante:

[↑prod(v, i), ↓prod(v, i)] **before** [↑emis(v, i), ↓emis(v, i)]
[↑emis(v, i), ↓emis(v, i)] **overlaps** [↑rec(v, i), ↓rec(v, i)]
[↑rec(v, i), ↓rec(v, i)] **before** [↑cons(v, i), ↓cons(v, i)]

De même, l'enchaînement de FA glissantes représenté dans la figure 4.b peut se décrire de la manière suivante:

[↑prod(v, i), ↓prod(v, i)] **overlaps** [↑emis(v, i), ↓emis(v, i)]
[↑emis(v, i), ↓emis(v, i)] **overlaps** [↑rec(v, i), ↓rec(v, i)]
[↑rec(v, i), ↓rec(v, i)] **overlaps** [↑cons(v, i), ↓cons(v, i)]

Dans le cas de l'enchaînement des FA temporelles fixes, les règles R4 et R5 permettent respectivement d'indiquer que l'instant de fin de la fenêtre temporelle de production est toujours antérieur à l'instant de début de la FA d'émission et que l'instant de fin de la FA de réception est toujours antérieur à l'instant de début de la FA de consommation. Les règles R4 et R5 se décrivent de la manière suivante:

R4: ↓prod(v, i) < ↑emis(v, i)
R5: ↓rec(v, i) < ↑cons(v, i)

Les différents cas d'enchaînement de FA fixes et de FA glissantes peuvent se décrire de la manière suivante :

- un enchaînement de fenêtres glissantes = R1 + R2 + R3
- un enchaînement de fenêtres fixes = R1 + R2 + R3 + R4 + R5
- un enchaînement de fenêtres fixes chez le producteur et de fenêtres glissantes chez le consommateur = R1 + R2 + R3 + R4
- un enchaînement de fenêtres glissantes chez le producteur et de fenêtres fixes chez le consommateur = R1 + R2 + R3 + R5

Lorsqu'un début de FA, représenté par ↑y(v, i+1), est élaboré de manière locale ou distante, alors ce dernier devra toujours être supérieur au début de la FA précédente.
Ceci nous conduit à élaborer la règle suivante qui s'applique à la production, à l'émission, à la réception et à la consommation.

R6: ↑y(v, i+1) > ↑y(v, i) avec y=(prod, emis, rec ou cons)

4.3 Fenêtres temporelles de validité des variables

Une variable produite à un instant donné possède une durée de vie qui varie selon les consommateurs de cette dernière.
Ainsi dans le cas d'une application industrielle, pour un processus d'application qui fait des statistiques, une certaine donnée peut être considérée comme valide avec une durée infinie. Par contre, pour un processus d'application lié à un actionneur intervenant dans un environnement temps réel, cette même donnée aura, par exemple, uniquement une validité de 20 ms. Par conséquent, la notion de durée de vie d'une donnée dépend de l'utilisation qui en est faite.

Lors de la production d'une variable, cette dernière possède une durée de vie, qui peut être différente à chaque production. Si l'on souhaite que l'entité de consommation connaisse la durée de vie d'une variable produite, alors dans certains cas, il est nécessaire de transmettre cette durée de vie vers les entités de consommation en même temps que la valeur de la variable et que les statuts temporels indiquant si les contraintes de production et d'émission ont été respectées.

Pour représenter la durée de vie d'une donnée, on utilisera à nouveau le mécanisme de fenêtre temporelle.

4.3.1 Notations

Nous allons maintenant essayer de formaliser le concept de durée de vie d'une variable de la même manière que nous l'avons fait pour les opérations de production, d'émission, de réception et de consommation. Une fenêtre temporelle concernant la durée de vie (ou la validité) d'une variable sera notée **FV**.

Nous utiliserons les mêmes notations $\uparrow$, $\downarrow$ et Δ que celles utilisées par les FA.

Les paramètres **i** et **v** seront utilisés pour représenter respectivement la $i^{ème}$ FV, instanciée lors de la production d'une variable v dans une FA.
Une instance d'une variable ne sera conservée que dans le cas où cette dernière est susceptible d'être consommée par un consommateur.

Une FV est toujours liée à une FA donnée, puisque le début de la FV a toujours lieu dans une FA de production. De plus, il est important de noter qu'il est possible d'avoir plusieurs productions d'une même variable donnée dans une même FA de production.

Lorsqu'une variable est produite, elle possède une certaine durée de vie donnée Δ, et elle possède une durée de vie Δ' (avec $\Delta'<\Delta$) lorsqu'elle est reçue chez l'entité consommatrice.

Une fenêtre temporelle concernant la validité d'une variable se décrit de la manière suivante:

$$x\,val(v, i) \quad \text{avec } x = (\uparrow, \downarrow \text{ ou } \Delta)$$

Le début de la FV, noté $\uparrow val(v, i)$, représente l'instant où la variable v est produite ; la fin de la FV, noté $\downarrow val(v, i)$, représente l'instant de fin de validité de la variable v.

4.3.2 Règles d'enchaînement

Les différents paramètres d'une FV doivent généralement être définis par l'utilisateur.
Nous allons donc poser un certain nombre de règles qui vont permettre de lier les FA avec les FV.

Comme pour les FA, le début d'une fenêtre temporelle concernant la validité d'une variable est toujours antérieur à la fin de cette même fenêtre temporelle. Ceci se traduit par la règle suivante:

$$R7: \quad \uparrow val(v, i) < \downarrow val(v, i)$$

Nous avons poser que l'instant de production d'une variable correspond toujours au début de la FV, il doit donc obligatoirement se situer dans une FA de production puisque le but d'une FA de production est de vérifier que l'on a bien produit dans les délais. Ce cas s'exprime de la manière suivante:

$$R8: \quad \uparrow prod(v, i) \leq \uparrow val(v, i) \leq \downarrow prod(v, i)$$

Concernant la fin de la fenêtre temporelle de validité d'une variable, les deux principaux cas qui peuvent se présenter sont:

- la fin de la durée de vie d'une donnée intervient avant la fin de la FA de consommation,
- la fin de la durée de vie d'une donnée intervient après la fin de la FA de consommation.

Ainsi, suivant la durée de vie et l'instant de production de la variable, la fin de validité de la variable peut intervenir avant ou après la fin de la FA de consommation. La figure ci-dessous permet d'illustrer cette notion de durée de vie des données par rapport à la FA de consommation:

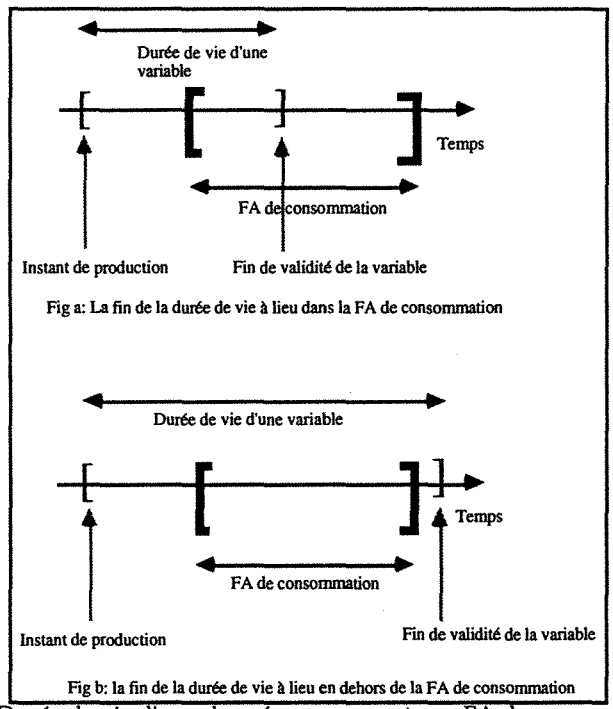


Fig. 5: Durée de vie d'une donnée par rapport aux FA de consommation

Le cas où la fin de la durée de vie d'une variable intervient avant la fin de la fenêtre temporelle de consommation se traduit par l'intermédiaire de la logique des intervalles de la manière suivante:

[↑val(v, i), ↓val(v, i)] **finishes** [↑prod(v, i), ↓cons(v, i)]

dont un cas particulier est:
[↑val(v, i), ↓val(v, i)] **during** [↑prod(v, i), ↓cons(v, i)]

Le cas où la fin de la durée de vie d'une variable intervient après la fin de la fenêtre temporelle de consommation se traduit par l'intermédiaire de la logique des intervalles de la manière suivante:

[↑prod(v, i), ↓cons(v, i)] **overlaps** [↑val(v, i), ↓val(v, i)]

En fonction de la durée de vie d'une donnée, il est possible d'émettre cette donnée plus ou moins tard, tout en émettant une donnée valide vers l'entité de consommation.
De même chez l'entité de consommation, en fonction de la durée de vie d'une donnée, il est possible de consommer cette donnée plus ou moins tard, tout en consommant une donnée valide.

Comme la durée de vie d'une variable doit permettre à l'entité de production ainsi qu'à l'entité de consommation de savoir à tout moment, si la valeur d'une certaine variable est valide ou non, il est nécessaire d'introduire deux cas:

- dans un premier cas, la durée de vie d'une variable n'est connue que de manière locale par chacune des entités de consommation. C'est alors uniquement ↓val(v, i) qui va permettre de savoir lorsqu'une certaine variable n'est plus valide. Il n'est alors pas possible de connaître de manière globale chez les entités de production et de consommation, la date de fin de validité d'une certaine variable.

- dans un second cas, la durée de vie d'une variable est fixée de manière globale une fois pour toutes lors de la production de cette dernière. C'est alors à l'entité de consommation de calculer la durée de validité restante lorsqu'elle reçoit la variable.
On se rend compte ici de l'importance que prend la synchronisation des horloges. Si les horloges locales du producteur et du consommateur ne sont pas synchronisées, alors la notion de durée de vie des variables ne peut pas être prise en compte durant la communication entre l'entité productrice et l'entité consommatrice. Dans le cas où une même variable est consommée par plusieurs entités consommatrices, si les différentes horloges locales ne sont pas synchronisées entre elles, alors la validité de cette variable risque d'arriver à échéance à des instants différents chez les différentes entités consommatrices.

4.4 Configuration des fenêtres temporelles

Par la suite, afin de ne pas répéter constamment (resp. de consommation), nous allons nous intéresser uniquement à l'étape de production. Cependant, notre raisonnement reste valide pour l'étape de consommation.

Si la production d'une valeur est périodique, alors on génère de manière cyclique une FA de production. Si la production d'une valeur est apériodique,

alors on ne possède aucune information concernant la date de production de la prochaine valeur.

Une fenêtre temporelle peut avoir une longueur fixe ou une longueur variable. Deux cas peuvent alors se présenter pour la production périodique et apériodique concernant les fenêtres temporelles de production:

- soit Δprod(v, i) est de longueur fixe,
- soit Δprod(v, i) est de longueur variable.

Dans le cas où les différentes étapes de la communication sont connues et figées d'avance par configuration, alors ces étapes sont généralement périodiques et se déroulent dans des intervalles de longueur fixes.

Les quatre combinaisons de FA possibles sont alors les suivantes :

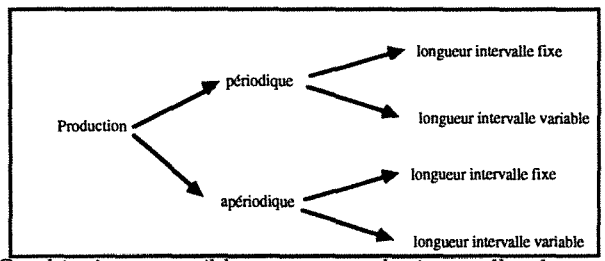


Fig. 6: Combinaisons possibles concernant les intervalles de production

Lors de l'étape de configuration, on peut fixer différents paramètres aux intervalles de production. Les quatre cas possibles sont les suivants :

- dans le premier cas (cf. figure 7.a), on souhaite que la production se fasse de manière périodique et que l'intervalle de temps alloué à la production soit de longueur fixe,
- dans le second cas (cf. figure 7.b), on souhaite que la production se fasse de manière périodique et que l'intervalle de temps alloué à la production soit de longueur variable,
- dans le troisième cas (cf. figure 7.c), on souhaite que la production se fasse de manière apériodique et que l'intervalle de temps alloué à la production soit de longueur fixe,
- dans le quatrième cas (cf. figure 7.d), on souhaite que la production se fasse de manière apériodique et que l'intervalle de temps alloué à la production soit de longueur variable.

Si on reprend le formalisme des FA, on obtient la représentation suivante :

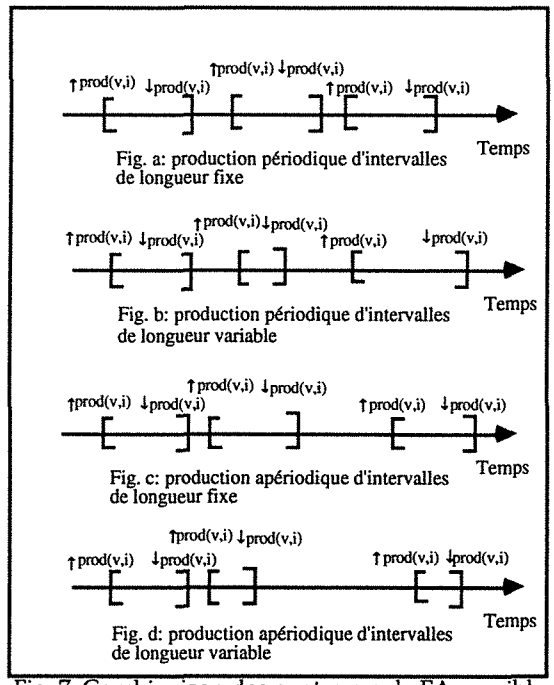


Fig. 7: Combinaison des quatre cas de FA possible

Notons que le précédent raisonnement appliqué aux FA peut être appliqué pour l'enchaînement des FV, qui peuvent également être générées de manière périodique ou apériodique et avec une longueur fixe ou une longueur variable.

Soulignons que l'on possède le maximum d'information lorsque les intervalles [↑y(v, i), ↓y(v, i)] et [↑val(v, i), ↓val(v, i)] apparaissent de manière périodique et avec une durée fixe. Par contre, on possède le minimum d'information lorsque les intervalles [↑y(v, i), ↓y(v, i)] et [↑val(v, i), ↓val(v, i)] apparaissent de manière apériodiques et avec une durée qui est variable. Rappelons que <y> = (prod, emis, rec ou cons).

Lorsque les FA et les FV apparaissent de manière périodiques et possèdent une longueur fixe, il est alors possible de faire transiter du trafic apériodique plus urgent à la place du trafic périodique tout en ayant des variables qui soient encore valides chez les entités consommatrices.

Dans un système temps réel, on est appelé à changer souvent la fréquence ainsi que la longueur des intervalles de temps des FA et des FV. Ainsi, par exemple, suite à l'arrivée d'un certain événement, il peut s'avérer nécessaire de reconfigurer les divers paramètres des fenêtres temporelles de façon à produire avec une vitesse différente.

Dans le cadre de la gestion de réseau [ISO 89a], l'étape de configuration a en charge vérifier la cohérence du système. Elle permet donc de détecter, par exemple, le cas où la production d'une donnée se fait de manière plus rapide que la consommation, chez les entités consommatrices, de cette même donnée ; ce qui surcharge de manière inutile le réseau de communication.
Les contraintes de temps de l'entité de production doivent donc prendre en compte les contraintes de temps des entités de consommation. De même les contraintes de temps des entités de consommation doivent prendre en compte les contraintes de temps des entités de production: il ne sert à rien de consommer plusieurs fois de suite une même valeur, si l'on sait d'avance que cette dernière n'a pas eu le temps d'être remise à jour.

La période séparant deux productions d'une même variable est encadrée par des intervalles de temps (notés MIN-p et MAX-p) qui représentent respectivement les délais minimum et maximum que l'on peut avoir entre deux productions successives, tout en respectant les contraintes de temps. Ce cas peut se représenter de la manière suivante :

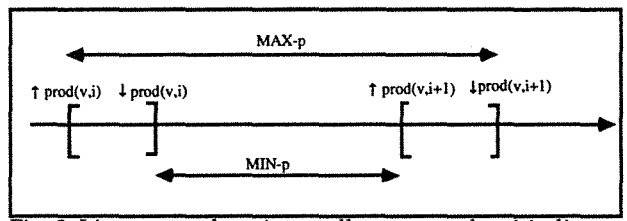


Fig. 8: Liens entre deux intervalles temporels périodiques

Durant l'intervalle de temps [↓prod(v, i), ↑prod(v, i+1)] qui correspond à MIN-p, il n'est pas possible de produire une valeur puisque on se situe en dehors des fenêtres temporelles de production. Ainsi, la période de consommation chez le consommateur ne doit pas être inférieure à la période minimale de production MIN-p, sinon on va consommer plusieurs fois la même valeur sans que cette dernière ait changé [RAJ 93], [DEC 93].
C'est au gestionnaire de réseau de détecter, lors de l'étape de configuration du réseau, les éventuelles incohérences temporelles énoncées lors de la spécification des besoins et de répondre au mieux aux exigences temporelles en fonction des priorités des divers intervalles de temps à un instant donné [LOR 90].

Pour élaborer les délais associés aux fenêtres temporelles, il est nécessaire de prendre en compte diverses contraintes : la périodicité des informations échangées, la vitesse maximale de transmission permise par le médium de communication utilisé, les services offerts par le réseau utilisé (notons que ces services dépendent du modèle de communication choisi).

Dans le cas du trafic apériodique, on ne possède généralement aucune information concernant les instants de production et de consommation.
Par contre dans le cas du trafic périodique, la connaissance des périodes de production et de consommation vont permettre de proposer un ensemble de règles permettant de lier les différentes fenêtres temporelles. C'est alors la période de production qui va conditionner l'ouverture de la prochaine fenêtre temporelle de production. Dans le cas d'une émission, d'une réception et d'une consommation périodique d'une variable, c'est la période de consommation qui va conditionner l'ouverture des prochaines fenêtres temporelles d'émission, de réception et de consommation.

L'étape de configuration est donc une étape très importante, puisqu'elle permet de vérifier que les différentes contraintes temporelles sont cohérentes entre elles et qu'elles permettent de bien répondre aux besoins énoncés.

5. Conclusion

Dans ce chapitre, nous avons étudié comment le temps pouvait se représenter dans les architectures de communication. Pour cela, nous avons introduit les concepts de fenêtres temporelles de production, d'émission, de réception et de consommation, ainsi que les concepts de fenêtres temporelles concernant la validité des variables afin de pouvoir formaliser les concepts de durée de vie d'une variable.

Nous avons proposé un ensemble de règles permettant d'enchaîner les différentes fenêtres temporelles de production, d'émission, de réception et de consommation entre elles, ainsi qu'avec les fenêtres temporelles concernant la validité des variables.
Lors de l'étape de gestion de réseau, on vérifiera que les paramètres respectent bien les différentes règles introduites.

Dans les chapitres suivants, nous essayerons de traduire par l'intermédiaire des réseaux de Nutt, ces mécanismes temporels en terme de machines d'états afin d'élaborer des statuts temporels.

Chapitre 3

Modélisation et validation des mécanismes de fenêtres temporelles

Chapitre 3

Modélisation et validation des mécanismes de fenêtres temporelles

1. Introduction

Dans le chapitre précédent, nous avons introduit les concepts de fenêtres temporelles d'activité de production, d'émission, de réception et de consommation, ainsi que les concepts de durée de validité des variables. Nous avons ensuite présenté les différentes relations existantes entre ces concepts. Nous proposons maintenant de modéliser les relations entre les fenêtres temporelles par l'intermédiaire d'une machine d'états utilisant les réseaux de Nutt [NUT 72].

Nous allons donc développer une machine d'états générale qui permet d'élaborer un ensemble de statuts temporels permettant de savoir, à un instant donné, si l'on se trouve ou non dans une fenêtre temporelle d'activité de production, d'émission, de réception ou de consommation et de connaître l'état de la validité des variables.
L'instanciation de la machine d'états générale va permettre de réaliser une fonction de contrôle des contraintes temporelles pour une fonctionnalité donnée (production, émission, réception ou consommation).
Nous validerons ensuite cette machine d'états générale par l'intermédiaire du logiciel AUTO et nous proposerons une simulation de cette machine d'états en utilisant TCP/IP.

2. Modélisation des fenêtres temporelles

Dans le chapitre précédent, nous avons introduit deux types de fenêtres temporelles, les FA et les FV.
A un instant donné, on doit être capable de savoir si on se situe à l'intérieur ou à l'extérieur d'une FA et d'une FV. Indépendamment du fait que l'on se situe ou non dans une FA, il peut être important de savoir s'il existe une $(i-1)^{ème}$ FV pour un variable v donnée. Si on considère tous les différents cas possibles, on obtient le tableau suivant ci-dessous:

Les symboles utilisés sont les suivants:
0 = la condition n'est pas vérifiée
1 = la condition est vérifiée.

Dans FA	**Dans FV**	**Il existe une FV pour une variable v donnée**	**Nom de l'état**
0	0	0	E-Faux
0	0	1	Expiration2
0	1	-	Expiration1 (FA)
1	0	0	Attente
1	0	1	Expiration1 (FV)
1	1	-	E-Vrai

Exploitons maintenant ce tableau :

- a) on se situe hors d'une FA, hors d'une FV et il n'existe pas de FV pour une variable v donnée, nous appellerons cet état E-Faux,
- b) on se situe hors d'une FA, hors d'une FV et il existe une FV pour une variable v donnée, nous appellerons cet état Expiration2,
- c) on se situe hors d'une FA et dans une FV, nous appellerons cet état Expiration1 de la FA,
- d) on se situe dans une FA, hors d'une FV et il n'existe pas de FV pour une variable v donnée, nous appellerons cet état Attente,
- e) on se situe dans une FA, hors d'une FV et il existe une FV pour une variable v donnée, nous appellerons cet état Expiration1 de la FV,
- f) on se situe dans une FA et dans une FV, nous appellerons cet état E-Vrai.

Si on représente les différents cas possibles en utilisant le formalisme des fenêtres temporelles, on obtient les représentations suivantes:

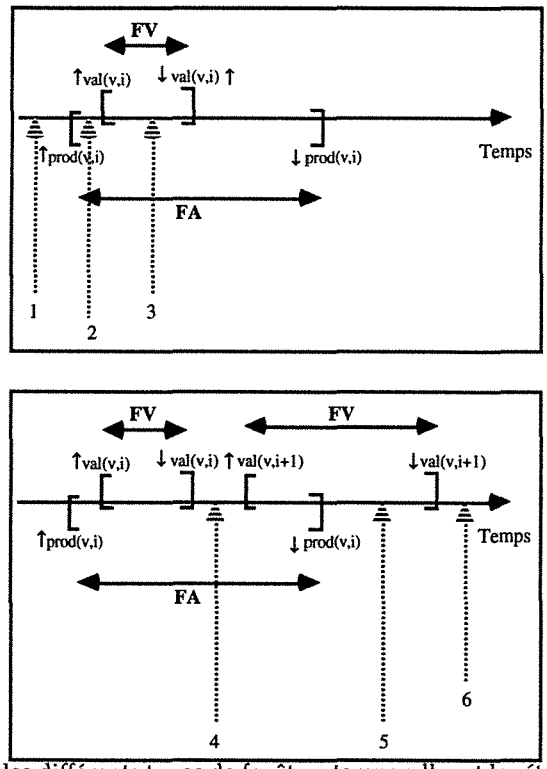


Fig. 1: Liens entre les différents types de fenêtres temporelles et les états correspondants

Explicitons chacun des six situations représentés:

- 1: on ne se situe ni dans l'intervalle délimité par [↑prod(v, i), ↓prod(v, i)], ni dans l'intervalle [↑val(v, i), ↓val(v, i)] et il n'existe aucune variable v produite dans une FV précédente qui soit disponible. Cet état correspond à l'état E-Faux.

- 2: on se trouve donc dans l'intervalle délimité par [↑prod(v, i), ↓prod(v, i)] sans qu'il n'y ait de variable v produite dans une FV précédente qui soit disponible. Cet état correspond à l'état Attente.

- 3: on se situe dans l'intervalle délimité par [↑prod(v, i), ↓prod(v, i)] et dans l'intervalle [↑val(v, i), ↓val(v, i)]. Cet état correspond à l'état E-Vrai, qui indique qu'au moins une production d'une variable v qui a été produite dans une FV précédente en respectant les contraintes de temps.

- 4: on se trouve dans l'intervalle délimité par [↑prod(v, i), ↓prod(v, i)], hors d'un intervalle [↑val(v, i+1), ↓val(v, i+1)] et il existe une variable v produite dans une FV précédente qui est disponible. Cet état correspond à l'état Expiration1 de la FV.

- 5: on ne se situe hors d'un intervalle [↑prod(v, i), ↓prod(v, i)] mais dans un intervalle [↑val(v, i), ↓val(v, i)]. Cet état correspond à l'état Expiration1 de la FA.

- 6: on ne se situe hors d'un intervalle [↑prod(v, i), ↓prod(v, i)], hors d'un intervalle [↑val(v, i), ↓val(v, i)] et il existe une variable v produite dans une FV précédente qui est disponible. Cet état correspond à l'état Expiration2.

Les états 4 et 5 correspondent à l'état Expiration1 qui peut être atteint suite à l'expiration de la FA ou de la FV.

Notons que deux temporisateurs différents doivent donc être associés aux FA et aux FV: un premier pour l'expiration de la FA et un second pour l'expiration de la FV.

2.1. Présentation des différentes sous-machines d'états

2.1.1 Les réseaux de Nutt

Dans ce paragraphe, nous allons montrer comment les réseaux de Nutt [NUT 72] permettent de modéliser les mécanismes de fenêtres temporelles.

Citons que de nombreux autres modèles peuvent être utilisé, comme par exemple les réseaux de Pétri qui constituent un outil de modélisation des systèmes discrets [REU 91], [MER 76], [RAM 74]. Il existe deux types de réseaux de Pétri où l'on retrouve des aspects temporels: les réseaux de Pétri temporisés et les réseaux de Pétri temporels. La différence entre les réseaux de Pétri temporisés et les réseaux de Pétri temporels repose sur le fait que les réseaux de Pétri temporisés associent à chaque transition une durée de tir, alors que les réseaux de Pétri temporels associent deux paramètres temporels t1 et t2 appelés respectivement date de tir au plus tôt et date de tir au plus tard [DIA 92].

Les réseaux de Nutt [NUT 72] sont beaucoup moins riches en sémantique que les réseaux de Pétri, mais ils sont toutefois plus lisibles que ces derniers. Les réseaux de Nutt vont permettre de construire des machines d'états indiquant, à un instant donné, dans quel état on se situe [SHA 92].

Les symboles utilisés par les réseaux de Nutt sont les suivants:

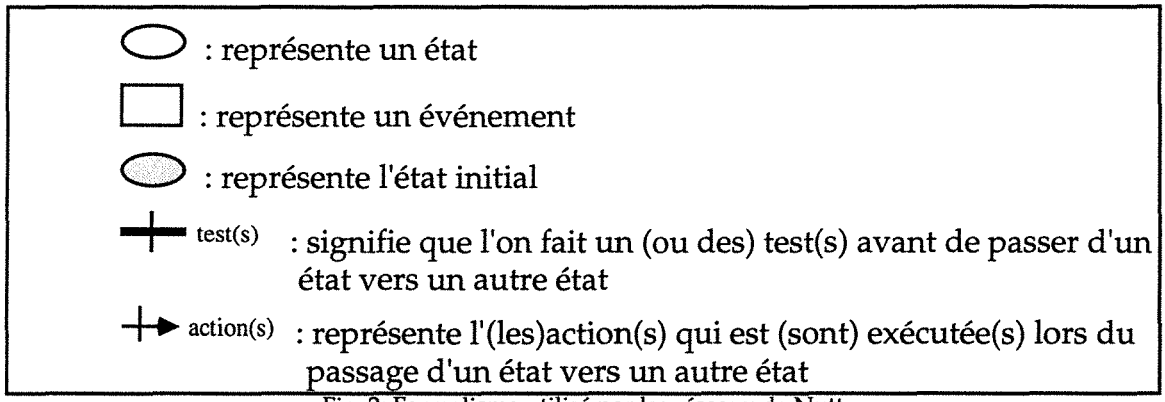


Fig. 2: Formalisme utilisé par les réseaux de Nutt

Sur réception d'un événement, on passe d'un état1 vers un état2 en exécutant une ou plusieurs actions. Dans le cas où il y a un test, on doit s'assurer que la (ou les) condition(s) est (sont) vérifiée(s) avant d'exécuter la (ou les) action(s) et de passer vers l'état2. Ce cas se représente de la manière suivante:

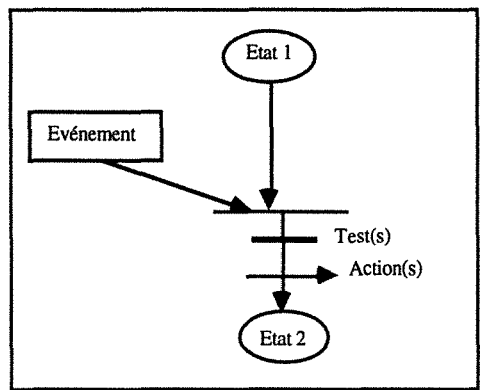


Fig. 3: Représentation du passage d'un état 1 vers un état 2 avec le formalisme des réseaux de Nutt

Nous utiliserons le formalisme proposé par les réseaux de Nutt, car ce dernier permet de représenter facilement un automate et le passage à l'étape de validation de l'automate par l'intermédiaire de l'outil AUTO [SIM 89], que nous utiliserons par la suite, se fait alors rapidement. De plus, comme la norme FIP [UTE 90a] utilise également le formalisme des réseaux de Nutt, la comparaison entre les machines d'états que nous allons introduire et les machines d'états du réseau de terrain FIP présentées dans le chapitre 4 sera facilitée.

Nous allons maintenant présenter les différentes sous-machines d'états permettant de passer d'un état vers un autre.
Signalons que nos machines d'états vont nous renseigner uniquement sur la validité de la dernière valeur produite, émise, reçue ou consommée.

2.1.2 Passage vers l'état Attente et de l'état Attente sur lui-même

Dès que l'on génère un ↑y(v, i+1) -noté ordre-de-début dans la machine d'états-, alors on recommence un nouveau cycle. Donc, afin de se différentier de la précédente FA et quelque soit l'état dans lequel on se trouve, on passe vers l'état Attente en armant un temporisateur lié à la nouvelle FA. Ce temporisateur va permettre de générer, par l'intermédiaire d'une machine d'états qui gère l'instant de fin de validité de la FA de production, un événement "expiration du temporisateur" correspondant à ↓y(v, i+1).
Notons que si l'on se situe déjà dans l'état Attente, alors lors de la réception d'un nouvel ordre-de-début, on reste dans ce dernier état mais en réinitialisant le temporisateur lié à la FA avec la nouvelle valeur reçue.

On peut représenter ce cas de figure de la manière suivante:

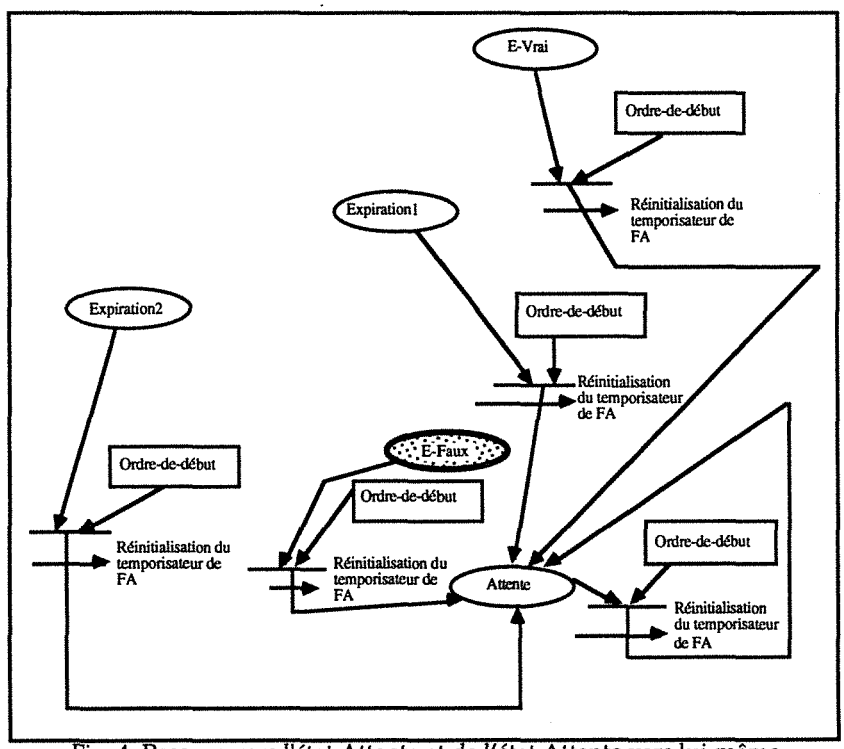


Fig. 4: Passage vers l'état Attente et de l'état Attente vers lui-même

2.1.3 Passage de l'état Attente vers l'état E-Faux

Si l'on se situe dans l'état Attente et si aucune production de variable n'a eu lieu, alors lorsqu'on reçoit une expiration du temporisateur matérialisée par un ↓y(v, i), on passe de l'état Attente vers l'état E-Faux.

Ceci ce schématise de la manière suivante:

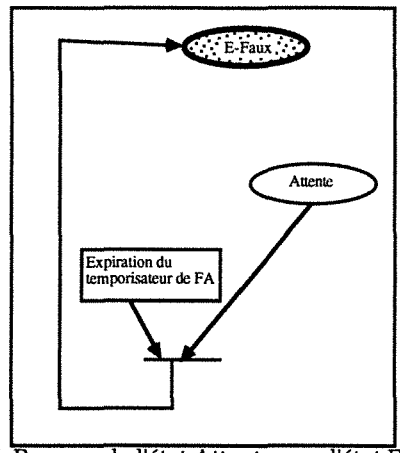


Fig. 5: Passage de l'état Attente vers l'état E-Faux

2.1.4 Passage de l'état Attente vers l'état E-Vrai et de l'état E-Vrai sur lui-même

Lorsque l'on se situe dans l'état Attente, la réception d'une nouvelle variable, fait passer vers l'état E-Vrai en armant un temporisateur lié à la validité de la variable.
Rappelons que la durée de validité est différente pour chacune des opérations de production, d'émission, de réception et de consommation.

Ainsi, lors de la production d'une nouvelle variable à l'instant ↑val(v, i+1), un temporisateur lié à la FV, permettant de générer un ↓val(v, i+1) est armé. Pour les entités d'émission, de réception et de consommation, le temporisateur de la FV est mis à jour en fonction de la durée de validité restante de la variable.
Notons que si l'on se situe déjà dans l'état E-Vrai, alors lors de la réception d'un nouvel ordre-de-début, on reste dans ce dernier état mais en mettant à jour le temporisateur lié à la FV avec la nouvelle valeur reçue.

Ce cas peut se représenter de la manière suivante:

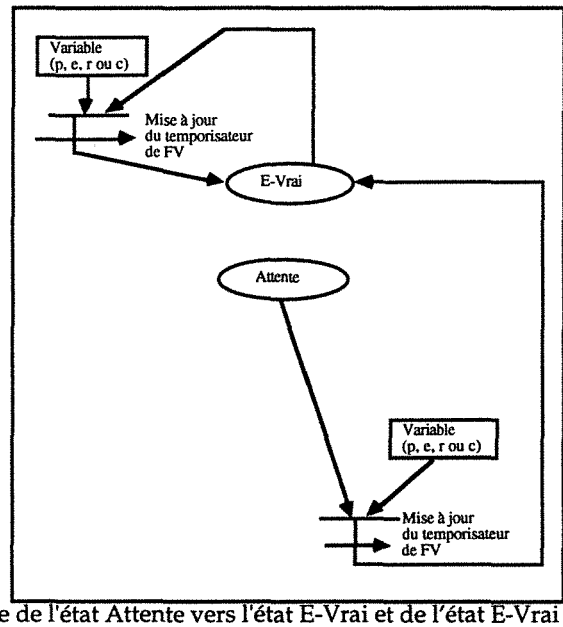


Fig. 6: Passage de l'état Attente vers l'état E-Vrai et de l'état E-Vrai vers lui-même

2.1.5 Passage de l'état E-Vrai vers l'état Expiration1

Suite à la réception d'une ↓y(v, i) ou d'une ↓val(v, i), on passe de l'état E-Vrai vers l'état Expiration1.
Lorsque l'on se situe dans l'état Expiration1 suite à une ↓val(v, i), si une nouvelle variable ↑val(v, i+1) est produite, alors on repasse à nouveau dans l'état E-Vrai en armant un temporisateur lié à la validité de la variable et permettant de générer par la suite un ↓val(v, i+1). Ce cas de figure est introduit par l'intermédiaire d'un test "première expiration = FV" entre le passage de l'état Expiration1 vers l'état E-Vrai.
Pour les entités d'émission, de réception et de consommation, le temporisateur de la FV est mis à jour en fonction de la durée de validité restante de la variable.

Les transitions entre ces deux états peuvent se représenter de la manière suivante:

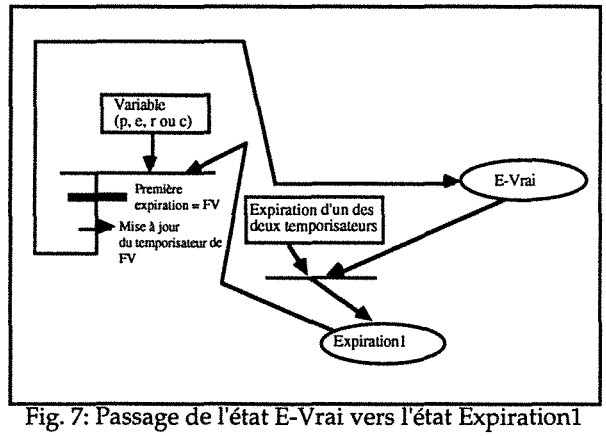


Fig. 7: Passage de l'état E-Vrai vers l'état Expiration1

2.1.6 Passage de l'état Expiration1 vers l'état Expiration2 et vers l'état E-Faux

Lorsque l'on se situe dans l'état Expiration1 suite à la réception d'un ↓y(v, i), si une variable ↑val(v, i+1) est produite, on passe alors de l'état Expiration1 vers l'état E-Faux puisque cette variable n'est pas produite dans une FA. Nous avons dû faire un test concernant le passage de l'état Expiration1 vers l'état E-Faux afin de ne pas prendre en compte le cas qui a été traité dans le paragraphe précédent. Par contre toujours dans le même état, si on reçoit un ↓val(v, i), alors on passe de l'état Expiration1 vers l'état Expiration2 pour indiquer que l'on ne se situe plus ni dans une FA, ni dans une FV, mais qu'il existe une variable v qui a été produite dans une $i^{ème}$ FV.

Lorsque l'on se situe dans l'état Expiration1 suite à la réception d'un ↓val(v, i), si un ↓y(v, i) est reçu, alors on passe de l'état Expiration1 vers l'état Expiration2 pour indiquer également que l'on ne se situe plus ni dans une FA, ni dans une FV, mais qu'il existe une variable v qui a été produite dans une $i^{ème}$ FV.

Si on se trouve dans l'état Expiration2, la réception d'une nouvelle variable ↑val(v, i+1) fait passer de l'état Expiration2 vers l'état E-Faux pour indiquer que la nouvelle variable n'a pas été produite dans une FA.

On représente alors les transitions de la manière suivante:

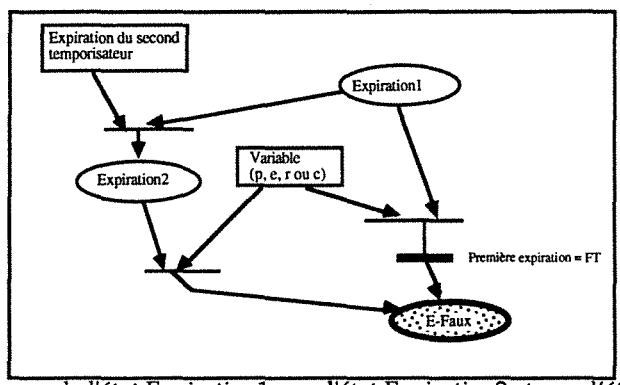


Fig. 8: Passage de l'état Expiration1 vers l'état Expiration2 et vers l'état E-Faux

2.2 Machine d'états générale

Dans le cas où on n'a pas de contraintes temporelles de date au plus tôt, mais uniquement des contraintes de date au plus tard, l'ordre-de-début ↑y(v, i) -introduit dans le paragraphe 2.1.2- qui permet de passer de l'état E-Faux vers l'état Attente est toujours mis à VRAI. On passe alors directement de l'état E-Faux vers l'état Attente, car dans ce cas cette notion d'ordre-de-début n'existe plus. Il est alors possible de produire à n'importe quel moment puisque dès qu'une expiration du temporisateur matérialisée par un ↓y(v, i) est reçue, on génère automatiquement un ↑y(v, i+1) qui fait passer vers l'état Attente.

Dans le cas où l'on n'a pas de contraintes temporelles de date au plus tard mais uniquement des contraintes temporelles de dates au plus tôt, alors lors de la génération d'un ordre-de-début ↑y(v, i), on arme le temporisateur de la FA avec une durée infinie. Ainsi lors de la production d'une variable, on reste dans l'état E-Vrai ou Expiration1 jusqu'au moment où l'on génère un nouvel ordre-de-début ↑y(v, i+1), qui permet de passer vers l'état Attente.

La machine d'états générale est donc conçue pour gérer des fenêtres temporelles comportant des contraintes temporelles de date au plus tôt et de date au plus tard. Mais elle convient également parfaitement bien aux fenêtres temporelles prenant en compte uniquement des contraintes de dates au plus tôt ou uniquement des contraintes de dates au plus tard [LOR 94a].

La machine d'états générale concaténant toutes les différentes sous-machines d'états introduites précédemment se représente de la manière suivante :

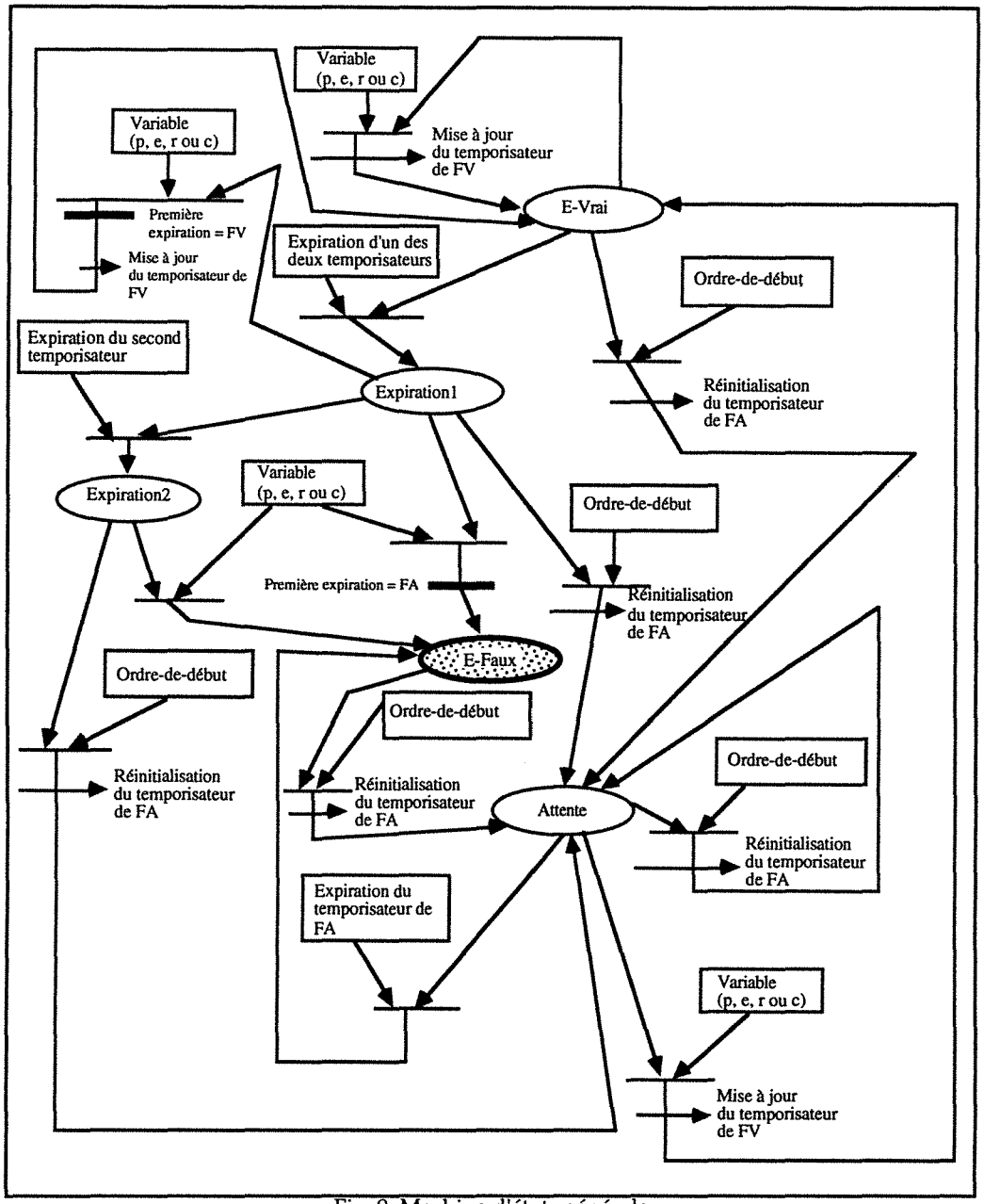


Fig. 9: Machine d'états générale

Il est nécessaire d'instancier et d'implémenter cette machine d'états générale à tous les niveaux de la communication pour élaborer des statuts temporels informant sur le respect des contraintes temporelles concernant la production, l'émission, la réception, la consommation ainsi que la validité des variables.

3. Validation et implantation de la machine d'états générale

3.1 Etape de validation

La vérification d'un protocole porte sur deux aspects: une vérification "sémantique" concernant la réalisation des services offerts par une couche (N) et une vérification "syntaxique" concernant la conformité des règles d'échange entre des entités de niveau (N) [BEL 88]. Pour effectuer cette vérification, on peut soit faire de la preuve théorique à partir d'une modélisation du système à vérifier, en faisant un parcours de tous les états possibles du graphe ou en s'appuyant sur la déduction logique des règles d'inférence ; soit faire de la preuve par simulation à partir d'une spécification formelle du protocole, on parle alors d'une vérification de la conformité vis-à-vis d'une spécification.

Différents outils peuvent être utilisés pour mettre en oeuvre l'étape de validation: citons l'outil de validation de protocole Mec [CRU 89] qui s'appuie sur le modèle Arnold-Nivat [ARN 89] et [NIV 82] et qui permet de manipuler des systèmes de transitions représentant des processus. Citons également l'outil AUTO [SIM 89], [ROY 89], [BOU 89] qui permet de faire de la preuve théorique et qui met en oeuvre une méthode de vérification basée sur la bissimulation en utilisant le calcul de processus Meije [BOU 85]. Nous avons choisi de retenir l'outil AUTO parce que ce dernier permet assez rapidement de mettre en oeuvre l'étape de validation à partir du formalisme des réseaux de Nutt que nous avons utilisé.

La méthode utilisée par le logiciel AUTO, développé par l'INRIA, pour vérifier la cohérence des différentes machines d'états consiste à réduire, par application de la bissimulation généralisée, la modélisation des différentes machines d'états. Ceci permet alors de restreindre l'analyse du système à une partie du système. L'utilisateur peut effectuer différents types de restrictions de son système et ainsi observer les différents comportements lorsque ces derniers seront suffisamment réduits.
Notons qu'il est possible d'orienter la validation, suivant les types de restriction mis en oeuvre.

Pour implanter la machine d'états générale introduite dans le paragraphe précédent en utilisant le logiciel AUTO, il est nécessaire d'introduire d'autres machines d'états afin de pouvoir générer les événements nécessaires à la machine d'états générale pour passer d'un état vers un autre. On décrira donc:

- la machine d'états *Producteur* ou *Consommateur* dont le but est de produire des ordres-de-début ainsi que de produire ou de consommer des variables,
- la machine d'états *Temporisateur* permettant d'envoyer un événement "expiration du temporisateur" suite à la réception d'un événement "armement

du temporisateur". Après un temps ∂t suite à la réception d'un événement *Armement du temporisateur,* la machine d'états *Temporisateur* renvoie un événement *Expiration du temporisateur* à la machine d'états qui lui a envoyé l'événement *Armement du temporisateur.* Pour chaque événement *Armement du temporisateur,* même si le temporisateur est sur le point de produire un événement *Expiration du temporisateur,* on réarme la fonction *Temporisateur* avec la dernière valeur reçue.

Une représentation de la modélisation des trois automates:
- producteur ou consommateur,
- machine d'états générale et
- temporisateur;

ainsi que des événements:
- production ou consommation d'une variable,
- ordre-de-début,
- armement et expiration du temporisateur de la FA et de la FV

implantés sur station SUN est la suivante:

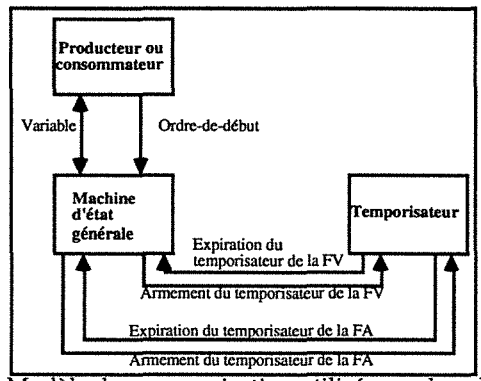


Fig. 10: Modèle de communication utilisé pour la validation

L'apport principal de l'outil AUTO est la souplesse qu'offre ce dernier pour passer d'une machine d'états, à la construction d'un automate qui permet alors de répondre à un certain nombre de besoins de validation (vérification des cycles, de la vivacité de l'automate, etc ...).

Afin de permettre la communication entre les trois entités décrites précédemment, il est nécessaire d'utiliser un tampon permettant à chaque machine d'états d'envoyer ou de recevoir des événements. L'automate "tampon" permet donc de modéliser la voie de communication entre les différentes machines d'états.
L'automate "tampon" peut être à une ou deux places, puisqu'il est possible, par exemple, de recevoir en même temps deux armements et deux expirations des temporisateurs pour la FA et pour la FV. De même, on peut avoir en même temps une variable et un ordre-de-début qui peuvent être générés.
Cet automate peut se décrire de la manière suivante dans le langage AUTO:

```
@ parse tampon=
meije0> let rec {vide= messrecep?: plein
meije0> + messrecep?:messrecep1?:plein1
meije0> and
meije0> plein= messenvoi!: vide
meije0> and
meije0> plein1= messenvoi!:messenvoi1!:vide}
meije0> in vide;
```

L'automate de l'entité producteur que nous appelons "genproduct" a en charge de produire les événements ordres-de-début et/ou les variables, peut s'écrire de la manière suivante:

```
@ parse genproduct=
meije0> let rec {pro= debut!:variable!:pro
meije0> + debut!:pro
meije0> + variable!:pro}
meije0> in pro;
```

L'automate de l'entité temporisateur pour la FA que nous appelons "gentemps1" se décrit de la manière suivante:

```
@ parse gentemps1=
meije0> let rec {repos= arme1?:armer1
meije0> and
meije0> armer1= expi1!:repos
meije0> + arme1?:armer1}
meije0> in repos;
```

L'automate de l'entité temporisateur pour la FV que nous appelons "gentemps2" se décrit de la manière suivante:

```
@ parse gentemps2=
meije0> let rec {repos= arme2?:armer2
meije0> and
meije0> armer2= expi2!:repos
meije0> + arme2?:armer2}
meije0> in repos;
```

L'automate de la machine d'états générale que nous appelons "general" s'écrit de la manière suivante:

```
@ parse general=
meije0> let rec { faux=debut?:arme1!:wait
meije0> and
meije0> wait=debut?:arme1!:wait
meije0> + expi1?:faux
meije0> + variable?:arme2!:vrai
meije0> and
meije0> vrai=expi2?:expivar
meije0> + variable?:arme2!:vrai
meije0> + debut?:arme1!:wait
meije0> + expi1?:expiwin
meije0> and
meije0> expivar=variable?:arme2!:vrai
meije0> + expi1?:expigen
meije0> + debut?:arme1!:wait
meije0> and
meije0> expiwin=debut?:arme1!:wait
meije0> + expi2?:expigen
meije0> + variable?:faux
meije0> and
meije0> expigen=variable?:faux
meije0> + debut?:arme1!:wait}
meije0> in faux;
```

L'automate partiel "genvoie" est le résultat de la composition de l'automate "general" et de l'automate "tampon", intervenant dans la modélisation de la communication des quatre événements consommés par l'automate "general".

```
@ parse genvoie=
meije0> ((((((general//tampon[variable/messenvoi, debut/messenvoi1])
meije0> \variable\debut))[variable/messrecep, debut/messrecep1])//
meije0> tampon [expi1/messenvoi][expi2/messenvoi1])
meije0> \expi1\expi2) [expi1/messrecep, expi2/messrecep1]);
```

L'automate partiel "genvoie", généré par l'outil Auto, est un automate obtenu par réduction par l'équivalence observationnelle avec 9 états, 27 transitions et 8 actions.

L'automate partiel "gentempvoie" est le résultat de la composition des automates "gentemps1", "gentemps2" et de l'automate "tampon", intervenant dans la modélisation de la communication des deux événements consommés par les automates "gentemps1" et "gentemps2".

@ parse gentempvoie=
meije0> (((gentemps1//gentemps2//tampon[arme1/messenvoi, arme2/messenvoi1])
meije0> \arme1\arme2)[arme1/messrecep, arme2/messrecep1]);

L'automate partiel "gentempvoie" est un automate obtenu par réduction par l'équivalence observationnelle avec 14 états, 58 transitions et 12 actions.

Enfin, l'automate "genaut" est le résultat de la composition des automates partiels "genvoie", "gentempvoie" et "genproduct".

@ parse genaut=
meije0> (gentempvoie//genvoie//genproduct)
meije0> \variable\debut\expi1\expi2\arme1\arme2;

L'automate "genaut" est un automate obtenu par réduction par l'équivalence observationnelle a un seul état qui valide donc le comportement de tous les autres automates.

Notons que, quel que soit l'ordre des regroupements des automates, on obtient toujours grâce à la propriété de congruence de l'équivalence observationnelle de Milner [MIL 83], le même automate final.

3.2 Implantation de la machine d'états générale

Nous avons implanté la machine d'états générale afin d'élaborer des statuts temporels permettant de savoir si la communication entre plusieurs machines via TCP/IP s'est faite en respectant les contraintes de temps fixées. Nous avons utilisé pour cela, des mécanismes de socket entre les différentes machines connectées sur le réseau.

Dans notre implantation, nous introduisons des contraintes temporelles de date au plus tard pour les FA, ainsi que des durées de validité pour les variables échangées. Pour les entités de production et de consommation, les différents statuts utilisés et présentés dans le paragraphe 2, sont les suivants:

1: on se situe dans l'état E-FAUX,
2: on se situe dans l'état Attente
3: on se situe dans l'état E-VRAI,
4: on se situe dans l'état Expiration1 de la FV,
5: on se situe dans l'état Expiration1 de la FA,
6: on se situe dans l'état Expiration2.

Dans l'exemple étudié, nous utilisons une station productrice et des stations consommatrices.

La station productrice envoie de manière périodique l'heure donnée par son horloge locale vers des stations consommatrices qui vont consommer ces valeurs avec une période qui leur est propre.
Nous allons poser que la durée de validité des variables produite est de **0,5 seconde** à partir de l'instant où elles ont été produites. Le statut de production est élaboré par l'entité de production et il est transmis en même temps que les données utiles vers les entités de consommation. Les statuts de consommation vont indiquer si les valeurs produites ont été consommées dans les délais.

Des statuts temporels seront élaborés, chez les entités productrices ainsi que chez les entités consommatrices, toutes les **0,25 seconde**.

Les statuts temporels chez l'entité de production vont permettre de savoir, à partir de l'échantillonnage de valeurs produites, quel est le pourcentage de valeurs produites qui ont respecté les contraintes de temps fixées par l'entité de production. Ainsi, en fonction de la charge de la machine, les contraintes de temps seront ou non respectées.

De même les statuts temporels chez l'entité de consommation vont permettre de savoir quel est le pourcentage de valeurs produites et consommées qui ont respecté les contraintes de temps fixées. Chez les entités de consommation les résultats obtenus vont dépendre fortement de la charge de la machine consommatrice ainsi que de la charge du réseau.
Pour chaque simulation, nous indiquerons donc la charge de la machine pour l'entité productrice et pour les entités consommatrices, ainsi que la charge du réseau durant la transmission des données de l'entité productrice vers les entités consommatrices.

Dans le cas où un producteur produit cinq variables toutes les 1 seconde, alors un exemple de simulation de l'entité de production avec une validité des variables égale à 0,5 seconde est le suivant:

```
Entrez le nombre de productions souhaitées: 5
Quelle est la fréquence de production souhaitée : 1
Produit : 15:11:11
   Statut de production : 1
   Statut de production : 1
   Statut de production : 1
   Statut de production : 1
Produit : 15:11:12
   Statut de production : 3
   Statut de production : 3
   Statut de production : 4
   Statut de production : 4
Produit : 15:11:13
   Statut de production : 3
   Statut de production : 3
   Statut de production : 4
   Statut de production : 4
Produit : 15:11:14
   Statut de production : 3
   Statut de production : 3
   Statut de production : 4
   Statut de production : 4
Produit : 15:11:17
   Statut de production : 1
   Statut de production : 1
   Statut de production : 1
   Statut de production : 1
Le pourcentage de requêtes produites respectant les contraintes de temps est de 60 %
La charge de la machine a été de 30 %
La charge du réseau a été de 9 %
```

Fig. 11: Exemple de simulation de l'entité de production

Dans cet exemple, on a produit les cinq valeurs suivantes: 15:11:11, 15:11:12, 15:11:13, 15:11:14 et 15:11:17.
Les valeurs 15:11:11 et 15:11:17 ont leur statut de production égal à 1 parce qu'elles n'ont pas été produites dans les délais.
Par contre, les valeurs 15:11:12, 15:11:13 et 15:11:14 ont été produites en respectant les contraintes de temps, donc leur statut est égal à 3. Ensuite, lorsque ces variables ne sont plus valides, leur statut prend la valeur 4 pour indiquer que la valeur n'est plus valide, mais que cette dernière a été produite dans les délais.

Toujours dans l'exemple ci-dessus, on peut constater qu'entre la quatrième et la cinquième production, la charge de la machine a été tellement importante, que la valeur "15:11:16" n'a pas été produite.

Dans cet exemple de simulation, la charge de la machine durant toute l'étape de simulation a été de 30 % ; la charge du réseau durant la transmission des données produites vers les entités consommatrices a été de 9 %.

Si l'on pose que le consommateur consomme toutes les 2 secondes, alors un exemple de simulation d'une entité de consommation est le suivant:

```
Quelle est la fréquence de consommation souhaitée : 2
Consomme : 15:11:11
   Statut de production : 1   Statut de consommation : 3
   Statut de production : 1   Statut de consommation : 3
   Statut de production : 1   Statut de consommation : 4
   Statut de production : 1   Statut de consommation : 4
   Statut de production : 1   Statut de consommation : 4
   Statut de production : 1   Statut de consommation : 4
   Statut de production : 1   Statut de consommation : 4
   Statut de production : 1   Statut de consommation : 4
Consomme : 15:11:13
   Statut de production : 3   Statut de consommation : 3
   Statut de production : 3   Statut de consommation : 3
   Statut de production : 4   Statut de consommation : 4
   Statut de production : 4   Statut de consommation : 4
   Statut de production : 4   Statut de consommation : 4
   Statut de production : 4   Statut de consommation : 4
   Statut de production : 4   Statut de consommation : 4
   Statut de production : 4   Statut de consommation : 4
Consomme : 15:11:17
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
   Statut de production : 1   Statut de consommation : 1
Le pourcentage de requêtes produites respectant les contraintes de temps est de 33 %
Le pourcentage de requêtes consommées respectant les contraintes de temps est de 67%
La charge de la machine a été de 40 %
```

Fig. 12: Exemple de simulation de l'entité de consommation

Dans cette seconde simulation, on a consommé trois valeurs: 15:11:11, 15:11:13 et 15:11:17.
Les valeurs 15:11:11 et 15:11:17 ont leur statut de production égal à 1 parce que ces valeurs n'ont pas été produites dans les délais (cf. fig. 11). Par contre, la valeur 15:11:13 a été produite dans les délais, donc le statut de cette dernière valeur est égal à 3, puis égal à 4 lorsque cette valeur n'est plus valide.

Les valeurs 15:11:11 et 15:11:13 ont leur statut de consommation égal à 3 parce que ces dernières valeurs ont été consommées dans les délais. Lorsque ces valeurs consommées ne sont plus valides, le statut devient égal à 4 pour indiquer que les valeurs ont été consommées dans les délais mais qu'elles ne

sont plus valides. La valeur 11:11:17 a son statut de consommation égal à 1 parce qu'elle n'a pas été consommée dans les délais.

La charge de l'entité consommatrice durant toute l'étape de simulation a été de 40 %.

Nous avons exécuté plusieurs fois de suite un même exemple, mais en faisant varier les charges de différentes machines et en faisant varier également la charge du réseau.

Dans le cas où l'on fait varier la charge de la machine et si la charge du réseau est constante, on obtient un pourcentage de requêtes produites respectant les contraintes de temps qui peut se représenter de la manière suivante:

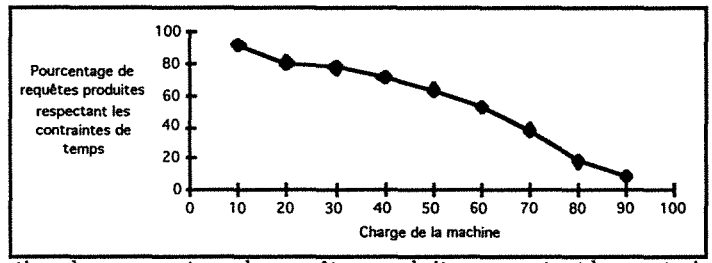


Fig. 13: Evolution du pourcentage de requêtes produites respectant les contraintes de temps en fonction de la charge de la machine

De même, si l'on fait varier la charge de la machine et si la charge du réseau est constante, on obtient alors un pourcentage de requêtes consommées respectant les contraintes de temps qui peut se représenter de la manière suivante:

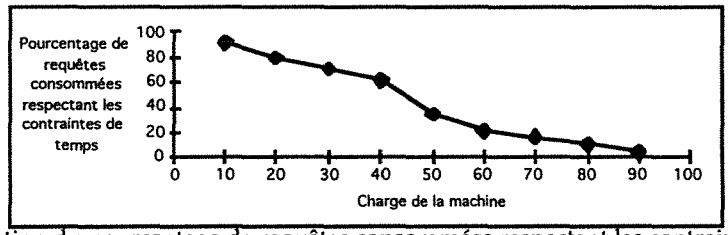


Fig. 14: Evolution du pourcentage de requêtes consommées respectant les contraintes de temps en fonction de la charge de la machine

On constate alors que, chez les entités de production et de consommation, plus la charge de la machine est élevée, plus le nombre de requêtes qui ont respectées les contraintes de temps est faibles.

Si l'on fait varier cette fois-ci la charge du réseau et si la charge de la machine consommatrice est constante, alors on obtient un pourcentage de requêtes consommées respectant les contraintes de temps qui peut se représenter de la manière suivante:

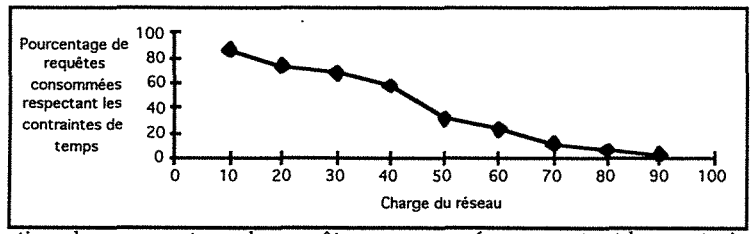


Fig. 15: Evolution du pourcentage de requêtes consommées respectant les contraintes de temps en fonction de la charge du réseau

On constate chez les entités consommatrices pour une charge de la machine qui reste constante, que plus la charge du réseau est élevée, plus le nombre de requêtes consommées respectant les contraintes de temps est faibles.

4. Application des mécanismes temporels étudiés aux modèles de communication

Nous avons présenté une étude des mécanismes temporels pour le modèle de communication producteur(s)/consommateur(s). Montrons maintenant que notre étude reste valide et peut se généraliser à d'autres modèles de communication tels que pour le modèle de communication client(s)/serveur(s).

Nous allons essayer de montrer que la machine d'états générale introduite précédemment, permet de vérifier le respect des contraintes temporelles de communication (production, émission, réception et consommation) et ceci aussi bien pour les modèles de communication producteur(s)/consommateur(s) que pour les modèles de communication client(s)/serveur(s).

4.1 Modèle Client(s)/Serveur(s) "temporel"

Dans le modèle de communication client/serveur, l'ordre de début-production introduit dans la machine d'états générale, correspond à la date à laquelle le client envoie sa demande vers le serveur. Lorsque le serveur reçoit une indication, il sait alors qu'il doit produire une certaine information à destination du client. Le rôle de l'ordre-de-début est donc d'initialiser un échange d'information entre deux entités communicantes.

Nous allons maintenant introduire des mécanismes temporels qui peuvent être utilisés dans le modèle de communication client/serveur, et ceci afin de savoir si la communication entre les entités a respectées les contraintes de temps fixées.

4.1.1 Mécanismes temporels de l'entité client

Les mécanismes temporels de l'entité client doivent permettre de savoir si, suite à l'envoi d'une requête à un serveur, le client a reçu la réponse du serveur dans les délais.
Entre le moment où une demande a été envoyée à un serveur par le client et la réception de la confirmation en provenance du serveur, le client se place dans l'état Attente après avoir armé un temporisateur. Si la confirmation de la requête n'arrive pas dans les délais souhaités, alors l'expiration du temporisateur fait passer le statut temporel de la machine d'états du client de l'état Attente vers l'état E-Faux. Si on se trouve dans l'état Attente, la réception de la requête de confirmation contenant l'information attendue permet à la machine d'états du client d'élaborer un statut temporel égal à vrai. L'envoi d'une nouvelle demande de la part du client fait passer le statut temporel vers l'état Attente après avoir armé un temporisateur. Lorsque l'on se situe dans l'état E-Vrai, l'expiration de la validité de la valeur de la variable reçue par le client, fait passer le statut temporel de la machine d'états de l'état E-Vrai vers l'état E-Faux.

La machine d'états temporelle du client peut se représenter de la manière suivante:

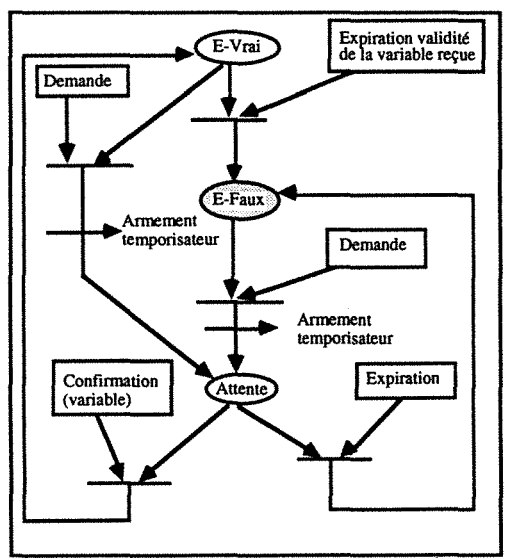


Fig. 16: Machine d'états temporelle pour le client

La machine d'états du client se déduit directement de la machine d'états générale. Il suffit alors simplement de fusionner dans la machine d'états générale, les états Expiration1 et Expiration2 avec l'état E-Vrai.

4.1.2 Mécanismes temporels de l'entité serveur

Les mécanismes temporels de l'entité serveur doivent permettre de savoir si le serveur a traité une certaine requête en provenance d'un client dans les délais. Entre le moment de la réception de l'indication en provenance du client et l'émission de la réponse (temps correspondant au temps de traitement du serveur), le serveur se place dans l'état Attente après avoir armé un temporisateur. Si le traitement ne se termine pas dans les délais, l'expiration du temporisateur fait passer le statut temporel de la machine d'états du serveur de l'état Attente vers l'état E-Faux. Si le serveur se trouve dans l'état Attente, alors l'émission de la requête de réponse contenant le résultat souhaité vers le client, permet d'élaborer un statut temporel égal à vrai. La réception d'une nouvelle indication fait passer le statut temporel vers l'état Attente après avoir armé un temporisateur. Lorsque l'on se situe dans l'état E-Vrai, l'expiration de la validité de la valeur de la variable émise par le serveur, fait passer le statut temporel de la machine d'états de l'état E-Vrai vers l'état E-Faux.
Notons que le statut élaboré par le serveur est envoyé vers le client en même temps que la valeur de la variable contenant le résultat du traitement.

La machine d'états temporelle du serveur peut se représenter de la manière suivante:

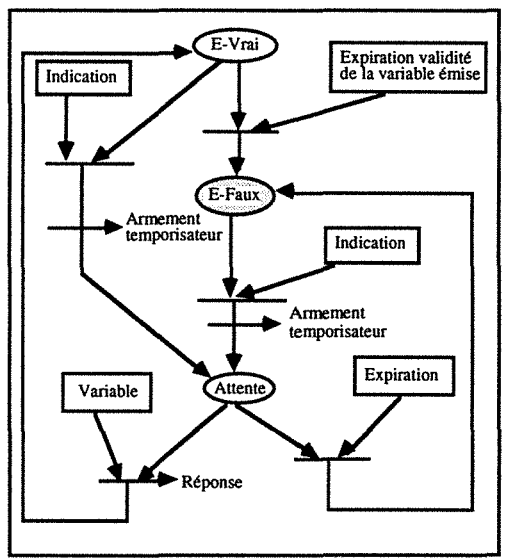


Fig. 17: Machine d'états temporelle pour le serveur

De la même manière que pour l'entité client, cette machine d'état se déduit directement de la machine d'états générale en fusionnant les états Expiration1 et Expiration2 avec l'état E-Vrai.

Remarquons que l'on peut valider les machines d'états temporelles du client et du serveur, directement à partir de la modélisation de la machine d'états

effectuée précédemment avec l'outil AUTO [SIM 89] ; et ceci par fusion des états Expiration1, Expiration2 et E-Vrai dans l'automate "general".
Nous avons fait ensuite une validation à partir du nouvel automate "general". Comme dans le cas de la validation faite dans le paragraphe 3.1, le nouvel automate "genaut" obtenu est un automate a un seul état qui valide alors le comportement de l'automate "general".

4.2 Liens entre les modèles de communication Client(s)/Serveur(s) et Producteur(s)/Consommateur(s)

L'envoi d'informations par le producteur (resp. le serveur) vers le ou les consommateurs (resp. le ou les clients dans le cas du modèle client/serveur) doit se faire dans une certaine fenêtre temporelle d'activité donnée, si l'on veut respecter les contraintes de temps et élaborer un statut temporel égal à vrai.

Une demande de service faite par un client dans le modèle de communication client/serveur ou une demande de production dans le modèle de communication producteur/consommateur correspond à l'envoi d'un événement ordre de début-production.

La réception de l'ordre de produire, aussi bien chez le producteur (dans le modèle producteur/consommateur) que chez le serveur (dans le modèle client/serveur), se matérialise dans la machine d'état générale par un ordre de début-production ↑prod(v, i), qui génère l'ouverture d'une FA de production. Dans les deux modèles de communication, la production (resp. l'émission, la réception et la consommation) devront se faire dans une FA de production (resp. d'émission, de réception et la consommation) si l'on souhaite élaborer un statut temporel égal à vrai.

5. Conclusion

Dans ce chapitre, nous avons poursuivi l'étude des aspects temporels dans la communication. Notre étude a permis de développer des mécanismes temporels permettant de répondre aux besoins de qualité de service dans la communication et elle nous a conduit à élaborer une machine d'états générale permettant de générer un ensemble de statuts temporels.

Ces statuts temporels nous informent sur la validité temporelle d'une certaine valeur ainsi que sur le respect des contraintes de temps lors des étapes de production, d'émission, de réception, de consommation.

L'utilisateur a bien sûr la possibilité d'implémenter tout ou une partie des mécanismes présentés. Cela dépend en fait du besoin de précision souhaité et des modèles de communication qui sont mis en oeuvre.
Nous avons ensuite validé la machine d'états générale et nous avons procédé à un ensemble de simulations.

Enfin, nous avons montré que la machine d'états introduite peut être utilisée aussi bien pour les modèles de communication producteur(s)/ consommateur(s) que pour les modèles de communication client(s)/serveur(s).

Chapitre 4

Application au réseau de terrain FIP

Chapitre 4

Application au réseau de terrain FIP

1. Introduction

Les mécanismes temporels présentés dans les chapitres précédents peuvent s'appliquer à divers types de réseaux de terrain. Dans ce chapitre, nous allons appliquer ces mécanismes au réseau de terrain FIP.

Dans la norme FIP [UTE 90a], on retrouve deux types de statuts temporels : le statut de rafraîchissement et le statut de promptitude. Trois caractères spécifiques sont associés à chacun de ces deux statuts: il s'agit du caractère synchrone, du caractère asynchrone et du caractère ponctuel.
Nous allons essayer de montrer comment la modélisation du temps introduite précédemment peut s'appliquer aux services de FIP qualifiés de temporel.

2. Le réseau de terrain FIP

FIP (Factory Instrumentation Protocol) [UTE 89], [UTE 90a], [UTE 90b], [UTE 90c], [UTE 92a], [UTE 92b] est un réseau de terrain ouvert avec une topologie en bus. Il peut être considéré comme un système de gestion de base de données industrielles réparties temps réel [THO 93].

Etude des différentes couches

Le réseau de terrain FIP ne retient que trois couches du modèle OSI [ISO 89a] : la couche physique, la couche liaison de données et la couche application.

- Les caractéristiques de la couche physique sont énoncées en annexe.

- La couche liaison de données de FIP est basée sur le modèle Producteurs/ Distributeur/Consommateurs (PDC). Une variable est produite par un seul producteur et est consommée par un ou plusieurs consommateurs. Le distributeur est responsable du déclenchement des transferts de données entre les producteurs et les consommateurs et a en charge d'assurer le respect des

contraintes temporelles. Un producteur est un processus d'application responsable de la production d'une donnée et le consommateur est un processus d'application qui a besoin d'une donnée pour s'exécuter [THO 90a].

Le modèle de communication Producteurs/Distributeur/Consommateurs peut se représenter de la manière suivante:

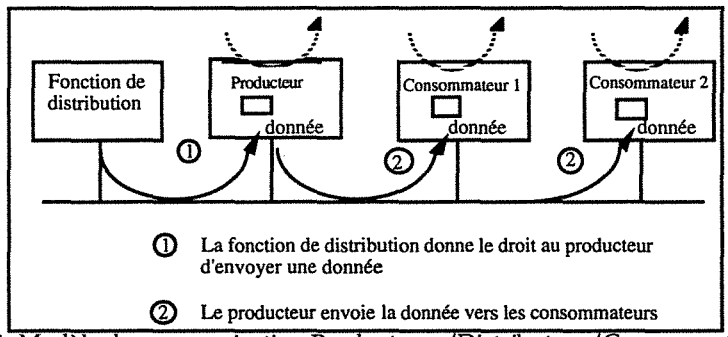


Fig. 1: Modèle de communication Producteurs/Distributeur/Consommateurs

Dans FIP, la gestion de l'accès au support de transmission se fait à partir d'une requête de la fonction de distribution qui permet au producteur d'identifier le nom de la variable dont la valeur doit être émise sur le réseau. Notons qu'aucune adresse physique de station n'est utilisée.

Le réseau FIP distingue le trafic périodique du trafic apériodique, puisque dans

- le trafic périodique c'est la fonction de distribution qui envoie de manière cyclique des ordres de production pour des variables données, par contre
- le trafic apériodique se déroule durant le temps libre restant après le trafic périodique qui a toujours priorité sur le trafic apériodique.

Deux types de trafics différents peuvent se présenter sur le réseau: le trafic de messages et le trafic de variables ; chacun de ces deux types de trafics pouvant être périodique ou apériodique. Les différences entre le trafic de messages et le trafic de variables proviennent du fait que:

- dans le trafic de messagerie:

  + les trames échangées sont de longueur variable,
  + on a la possibilité d'avoir des acquittements,
  + l'adresse du destinataire est contenue dans le message,
  + le temps de transfert du message n'est pas connu puisqu'il dépend de la position du message dans la file d'attente,
  + il n'y a pas de possibilité d'élaborer des statuts de cohérence temporelle et spatiale,

- par contre dans le trafic de variables:

  + la longueur des données échangées est variable mais toute trame échangée a une longueur fixe,
  + il n'y a pas d'acquittement,
  + les variables peuvent être diffusées à plusieurs consommateurs en même temps,
  + c'est au consommateur de s'arranger pour consommer l'information,
  + on peut élaborer des statuts de cohérences temporelle et spatiale.

Pour le trafic de variables du réseau de terrain FIP, les caractéristiques sont les suivantes:

+ 2^{16} objets identifiables au maximum,
+ échange cyclique de variables se fait en utilisant une table de configuration,
+ qualité de service: des statuts temporels permettent d'indiquer si les contraintes de temps énoncées concernant la mise à jour des variables ont été ou non respectées.

Pour le trafic de messagerie, les caractéristiques sont les suivantes:

+ point à point avec ou sans acquittement,
+ multipoints sans acquittement,
+ qualité de service: des temporisateurs permettent d'obtenir des délais bornés pour les échanges.

L'accès au support de transmission se fait par l'intermédiaire de la fonction de distribution qui est appelée arbitre de bus dans le réseau FIP. L'arbitre de bus diffuse les différents identifieurs en suivant une liste contenue dans un macro-cycle établi lors de l'étape de configuration [THO 91b].

Le temps pour exécuter un macro-cycle est borné. Ainsi, dans le cadre du trafic périodique, si la communication se fait correctement, une application est alors toujours en mesure de déterminer à quel instant précis une variable sera mise à disposition des utilisateurs.
Le macro-cycle est divisé en cycles élémentaires dont la durée est également fixée à l'avance lors de l'étape de configuration. Dans chaque cycle élémentaire, on retrouve une partie réservée pour le trafic périodique et s'il reste du temps, on a une partie allouée au trafic apériodique. Pour assurer une durée constante à chaque cycle élémentaire, on retrouve également une partie réservée à la synchronisation.

Un macro-cycle peut donc se schématiser de la manière suivante:

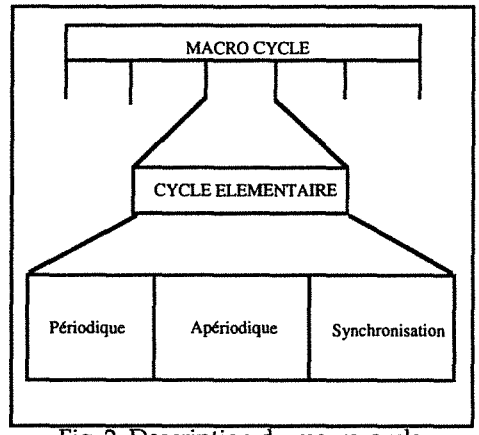


Fig. 2: Description du macro-cycle

Dans l'exemple ci-dessous, nous avons représenté un macro-cycle contenant 6 cycles élémentaires. Le trafic périodique est composé des variables suivantes:

- A produite toutes les 1 unité de temps,
- B produite toutes les 2 unités de temps,
- C et D produites toutes les 3 unités de temps,
- E et F produites toutes les 6 unités de temps.

Une méthode pour élaborer la table ci-dessous est développée dans [LOR 90], [CAR 93a] :

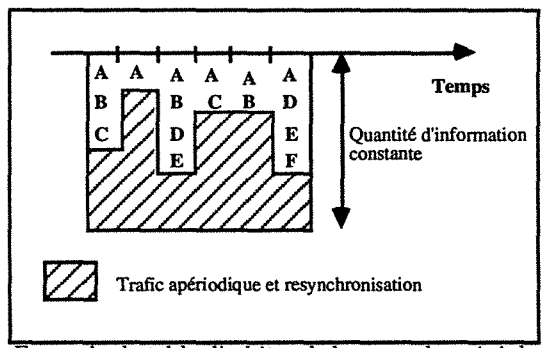


Fig. 3: Exemple de table d'arbitre de bus synchronisé dans FIP

Cet exemple permet d'illustrer le rôle essentiel joué par l'arbitre de bus afin d'assurer le respect des contraintes de temps lors de l'échange d'information entre les différentes stations.

- La couche application se compose de deux parties:

- la partie MPS (Manufacturing Periodic/Aperiodic Services) pour les services temps critique,
- un sous ensemble de MMS [ISO 89b] (Manufacturing Message Services) pour les services de messagerie.

La couche application du système de communication offre un ensemble de services qui permettent de lire et d'écrire les valeurs des variables. Ces variables peuvent être de type simple ou de type structuré et avoir différentes sémantiques (variables de synchronisation, variables de resynchronisation, ...). La couche application introduit deux catégories de services: une catégorie de services pour les opérations temps réel et une seconde pour les services de messagerie.

Dans MPS, on retrouve des services permettant de mettre en oeuvre de la:

- cohérence temporelle et spatiale,
- synchronisation et resynchronisation,
- sélection du cycle de scrutation en temps réel.

Les services de messagerie sont un sous-ensemble des services offerts par MMS et ils n'interfèrent pas avec les services temps réel.

Dans FIP deux types de statuts renseignent l'utilisateur sur la validité des données produites et consommées, il s'agit des statuts de promptitude et de rafraîchissement.

Le statut de rafraîchissement est élaboré par la couche application d'une entité productrice. Il permet d'informer les entités productrices et consommatrices, si la production d'une variable donnée s'est faite en respectant ou non les délais de production qui lui étaient alloués.

Le statut de promptitude est élaboré par la couche application d'une entité consommatrice. Il permet d'informer l'entité consommatrice si une variable donnée a bien été transmise ou non par le réseau dans les délais de transmission qui lui étaient allouées.

Ces mécanismes de promptitude et de rafraîchissement seront étudiés plus en détail par la suite.

Notons que le réseau de terrain FIP permet de renseigner les entités consommatrices sur le respects des contraintes de temps de production et d'acheminement des variables (**cohérences temporelle**) ainsi que sur l'identité des copies des variables (**cohérence spatiale**) chez toutes les différentes entités consommatrices.

• Concernant la gestion de réseau, les principaux services développés par le réseau de terrain FIP sont les suivants :

- définitions et mises à jour des listes d'objets et des tables de scrutation,
- gestion du démarrage et de l'arrêt des opérations,
- détection et correction des fautes.

La gestion de réseau joue un rôle très important dans FIP, puisque c'est elle qui a en charge de construire les tables de scrutation à partir des contraintes de temps énoncées par l'utilisateur.

L'organisation des différentes couches retenues par FIP peut se représenter de la manière suivante:

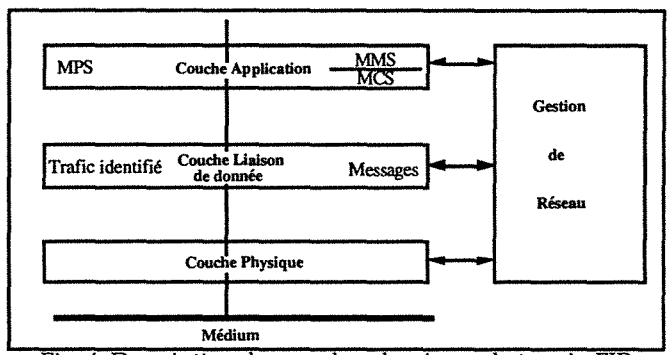


Fig. 4: Description des couches du réseau de terrain FIP

Le réseau de terrain FIP, par le choix du modèle de communication Producteurs/Distributeur/Consommateurs, de sa méthode d'accès au support de transmission au travers du mécanisme d'arbitre de bus et grâce aux services temporels introduits dans la couche application, permet d'offrir des qualités de services que l'on ne retrouve dans aucun des autres réseaux.

3. Statuts temporels pour le réseau de terrain FIP

Dans ce paragraphe, nous allons présenter tout d'abord les statuts temporels existants actuellement dans la norme FIP. Ensuite, nous proposons de compléter ces derniers en introduisant un nouveau statut temporel.

3.1 Présentation des statuts de rafraîchissement et de promptitude ainsi que de leurs caractéristiques

Nous allons présenter les deux statuts temporels présents dans le réseau de terrain FIP: le statut de rafraîchissement et le statut de promptitude.

3.1.1 Statut de rafraîchissement

Le concept de statut de rafraîchissement [UTE 90a] d'une variable est relatif à la validité temporelle concernant la **production** d'une donnée associée à une variable. Il permet de savoir si la valeur associée à une variable a été produite dans les délais alloués. Ce statut est donc élaboré par la station productrice de la variable.

Le statut de rafraîchissement permet de s'assurer que la production s'est faite dans les délais. Dans le cas contraire, ce statut indique que la station productrice est défaillante, que les contraintes de production allouées initialement à la configuration sont trop strictes ou encore que l'ordre de production n'est pas arrivé dans les délais souhaités.

Pour être le plus précis possible, ce statut doit être élaboré au plus haut niveau de l'architecture de communication du modèle OSI, c'est-à-dire au niveau de la couche application ou encore au niveau des normes d'accompagnement.

3.1.2 Statut de promptitude

Le concept de statut de promptitude [UTE 90a] d'une variable est relatif à la validité temporelle concernant la **transmission sur le médium** d'une donnée associée à une variable. Il permet de savoir si la valeur associée à une variable a été transmise sur le médium dans les délais alloués. Ce statut est élaboré par chacune des stations consommatrices de la variable, et ceci dès la réception de cette dernière. Un statut de promptitude possédant une valeur égale à faux, indique qu'il y a un problème de transmission, c'est-à-dire que les contraintes de transmission allouées initialement à la configuration sont trop strictes pour les performances permises par le réseau ou encore que l'ordre de transmission n'est pas arrivé dans les délais souhaités.

Pour être le plus précis possible, ce statut doit être élaboré au plus bas niveau de l'architecture de communication du modèle OSI, c'est-à-dire au niveau de la sous-couche MAC.

3.1.3 Caractéristiques des statuts de rafraîchissement et de promptitude

Dans la norme FIP, différents caractères ont été développés pour le rafraîchissement et pour la promptitude, il s'agit des caractères asynchrone, synchrone et ponctuel [UTE 90a]:

- le statut du caractère asynchrone est égal à vrai si la production de la valeur d'une variable (resp. la transmission) s'est faite depuis un temps

inférieur à la période de production (resp. de transmission) de cette même variable. On ne se préoccupe pas ici de l'instant de début, mais uniquement des contraintes de date au plus tard. Dans ce cas, les seules contraintes temporelles que l'on prend en compte concernent la fin de la fenêtre temporelle de production (resp. de transmission) puisque on cherche à savoir uniquement si on a produit (resp. transmis) avant l'expiration du temporisateur.

- le statut du caractère synchrone est égal à vrai si la production de la valeur d'une variable (resp. la transmission) s'est faite après un certain ordre de synchronisation et depuis un temps inférieur à la période de production (resp. de transmission). Un temporisateur est armé dès la réception de l'ordre de synchronisation. L'expiration de ce temporisateur intervient à la fin de la période de production (resp. de transmission). L'ordre de synchronisation permet de prendre en compte les contraintes de date au plus tôt et le mécanisme d'expiration de temporisateur permet de prendre en compte les contraintes de date au plus tard. Les contraintes temporelles qui sont prises en compte concernent donc le début et la fin de la fenêtre temporelle de production (resp. de transmission) puisque on cherche à savoir si on a produit une variable entre le moment où on a reçu un ordre de synchronisation et l'arrivé de l'expiration du temporisateur.

- le statut du caractère ponctuel est égal à vrai si la production de la valeur d'une variable (resp. la transmission) s'est faite au moins une fois après un ordre de synchronisation ; ce caractère reste égal à vrai tant qu'aucune nouvelle valeur pour une variable donnée n'est produite (resp. transmise) en dehors de la période de production (resp. de transmission). Les seules contraintes temporelles qui sont prises en compte concernent d'une part, le début de la fenêtre temporelle de production (resp. de transmission) puisque on cherche à savoir si on a produit une variable après avoir reçu un ordre de synchronisation et d'autre part, la fin de la fenêtre temporelle associée à la durée de vie de la variable car le statut change avec la fin de la validité de la variable produite (resp. transmise). Il suffit alors de poser que la fin de la validité de variable intervient dès la production d'une nouvelle valeur pour cette même variable.

Le caractère ponctuel complète donc l'information reçue par le caractère synchrone dans le sens où le caractère synchrone peut être égal à faux alors que le caractère ponctuel est encore égal à vrai: dans ce cas, on est sûr de ne plus être dans la fenêtre temporelle de production (resp. la transmission) mais que la valeur présente dans le buffer de l'entité productrice est une valeur qui a été produite dans les délais initialement alloués. Les caractères synchrones et ponctuels peuvent être tous les deux égaux à vrai ou tous les deux égaux à faux, mais on ne peut pas avoir le caractère synchrone égal à vrai et le caractère ponctuel égal à faux car si le caractère ponctuel est égal à faux, cela signifie déjà que l'on ne se situe plus dans la fenêtre temporelle de production (resp. la transmission) et alors le caractère synchrone est obligatoirement égal à faux.

Signalons que l'ordre de synchronisation introduit dans la norme FIP, correspond au début de la FA.

3.2 Introduction d'un nouveau statut temporel: le statut de fraîcheur-de-consommation

Pour couvrir l'ensemble du cycle de communication entre un producteur et des consommateurs (la production, la transmission et la consommation), nous avons introduit un statut qui ne se retrouve pas dans la norme FIP et que nous appelons statut de fraîcheur-de-consommation.

Le concept de statut de fraîcheur-de-consommation d'une variable est relatif à la validité temporelle concernant la **consommation** d'une donnée associée à une variable reçue. Il permet de savoir si la valeur associée à une variable a été consommée dans les délais alloués. Ce statut est donc élaboré par chacune des entités consommant une variable. Un statut de fraîcheur-de-consommation possédant une valeur égale à faux, indique qu'une variable donnée n'a pas été consommée dans les délais alloués ; par exemple, lorsque l'ordre de consommation n'est pas arrivé dans les délais souhaités ou encore lorsque aucune variable n'est disponible, etc ...

Pour être le plus précis possible, ce statut devra être élaboré au plus haut niveau de l'architecture de communication du modèle OSI, c'est-à-dire au niveau de la couche application ou encore au niveau des normes d'accompagnement.

3.3 Description des protocoles

Dans la norme FIP [UTE 90a], les statuts de rafraîchissement et de promptitude associés aux caractères asynchrones, synchrones et ponctuels sont représentés sous la forme de machine d'états décrites avec le formalisme des réseaux de Nutt [NUT 72].

Nous allons montrer que les caractères asynchrone, synchrone et ponctuel peuvent s'appliquer au statut de fraîcheur-de-consommation de consommation.

Si l'on applique les caractères asynchrones, synchrones et ponctuels au statut de fraîcheur-de-consommation introduit précédemment, on obtient les trois machines d'états suivantes que l'on a représenté avec le formalisme des réseaux de Nutt:

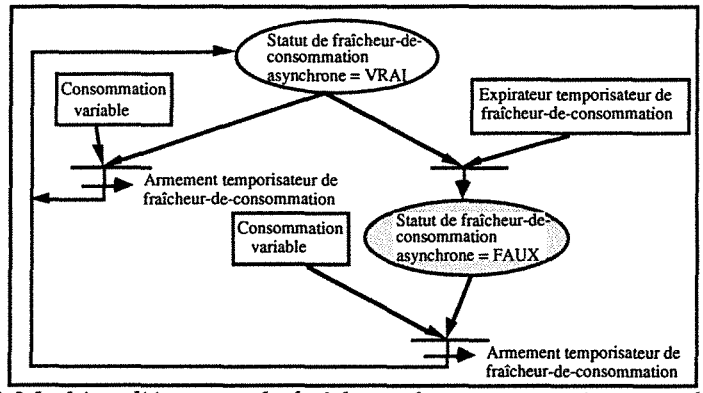


Fig. 5: Machine d'état pour la fraîcheur-de-consommation asynchrone

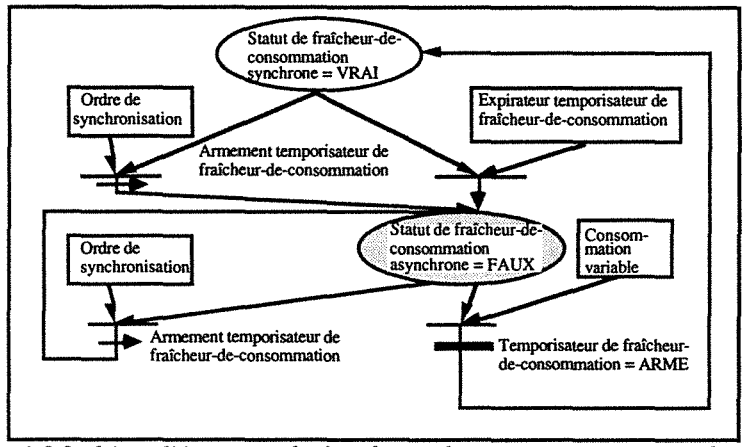


Fig. 6: Machine d'état pour la fraîcheur-de-consommation synchrone

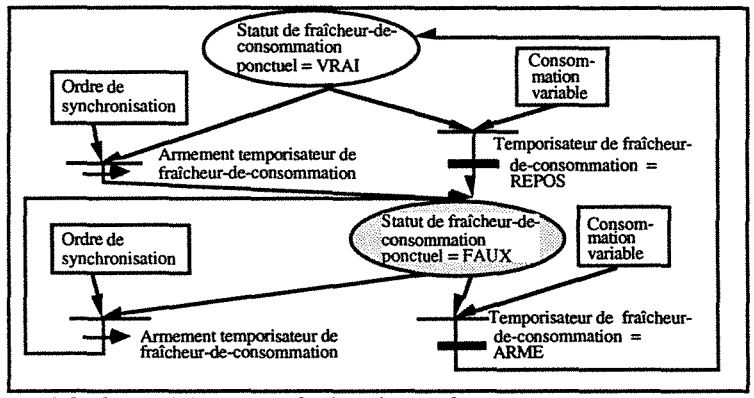


Fig. 7: Machine d'état pour la fraîcheur-de-consommation ponctuelle

Nous allons maintenant essayer de montrer comment à partir de notre étude des mécanismes temporels et de la machine d'états générale introduite dans les chapitres précédents, on peut retrouver les statuts temporels décrits dans le réseau de terrain FIP.

4. Application des mécanismes de fenêtres temporelles aux statuts temporels

4.1 Statuts temporels du réseau de terrain FIP

Quatre statuts temporels différents ont été introduits dans le chapitre 2:

- le statut temporel de production,
- le statut temporel d'émission,
- le statut temporel de réception,
- le statut temporel de consommation.

A travers ces statuts, on cherche à savoir si l'on a produit, émis, reçu ou consommé une certaine information dans les délais, c'est-à-dire dans une certaine FA donnée.

La figure suivante représente le cas où les opérations de production, d'émission, de réception et de consommation se sont déroulées dans les bonnes FA:

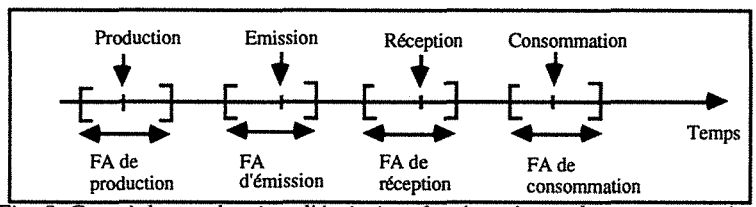


Fig. 8: Cas où la production, l'émission, la réception et la consommation de l'information ont bien respecté les délais alloués

Dans le réseau de terrain FIP, le statut de promptitude prend en compte le statut temporel d'émission et le statut temporel de réception puisque l'on teste si on a transmis une certaine variable dans une certaine fenêtre temporelle d'activité de transmission. Il est important de noter que dans le cas où les délais de transmission ne sont pas respectés, on ne sait pas si cela est dû au non respect des contraintes temporelles d'émission chez l'entité de production ou si cela provient du non respect des contraintes de temps lors de la transmission de l'information sur le médium de communication.

Le statut de rafraîchissement correspond au statut temporel de production et le statut temporel de fraîcheur-de-consommation correspond au statut temporel de consommation. Ainsi le statut de rafraîchissement (resp. de promptitude et de fraîcheur-de-consommation) indique si les contraintes temporelles (représentées sous forme de fenêtres temporelles) de production (resp. de transmission et de consommation) ont été respectées.
Si l'on conserve le même formalisme que celui introduit précédemment, les statuts de rafraîchissement et de promptitude introduits dans la norme FIP, ainsi que le statut de fraîcheur-de-consommation que nous avons introduit, peuvent se schématiser de la manière suivante:

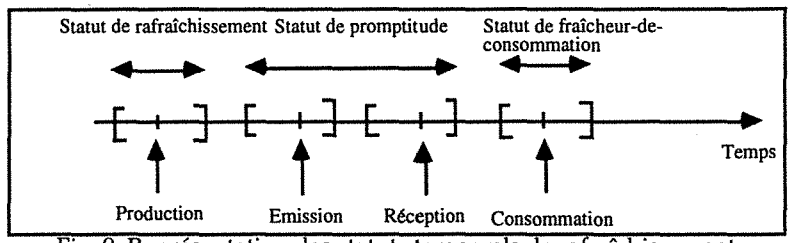


Fig. 9: Représentation des statuts temporels de rafraîchissement, de promptitude et de fraîcheur-de-consommation de FIP

Rappelons que dans le cas du réseau de terrain FIP, l'entité consommatrice reçoit de la part de l'entité productrice, en même temps que la valeur d'une certaine variable, un statut de rafraîchissement associé à cette variable qui indique si les contraintes temporelle de production ont été respectées. En complément du statut de rafraîchissement, l'entité consommatrice élabore, lors de la lecture de la variable reçue, un statut de promptitude [LOR 94b].

Essayons maintenant d'étudier et d'illustrer en détail ces différents mécanismes.

4.1.1 Statut de rafraîchissement

Le statut de rafraîchissement est égal à faux lorsque la fonction de production ne fonctionne pas correctement.
Soit t l'instant courant et soit une production qui se fait de manière périodique toutes les 10 unités de temps. Si par exemple, la valeur qui se trouve chez le producteur date de l'instant t-20, alors le statut de rafraîchissement est égal à faux, car on n'a pas produit dans les délais.

On peut représenter cet exemple de la manière suivante:

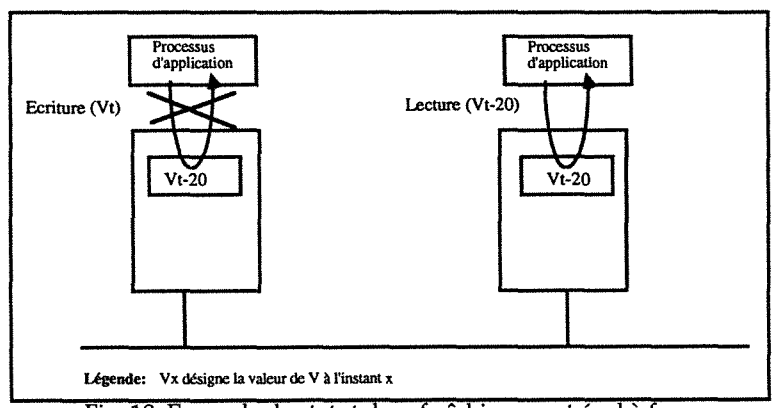


Fig. 10: Exemple de statut de rafraîchissement égal à faux

On peut noter que le statut de rafraîchissement est indépendant des autres statuts.

4.1.2 Statut de promptitude

Le statut de promptitude est égal à faux lorsque la fonction de distribution ne fonctionne pas correctement.
Soit le cas où une variable doit être transmise toutes les 10 unités de temps de l'entité productrice vers l'entité consommatrice. Si la valeur qui se trouve chez le producteur date de l'instant t et si la valeur qui se trouve chez le consommateur date de l'instant t-20, alors le statut de promptitude est égal à faux.

On peut représenter ce cas de figure de la manière suivante:

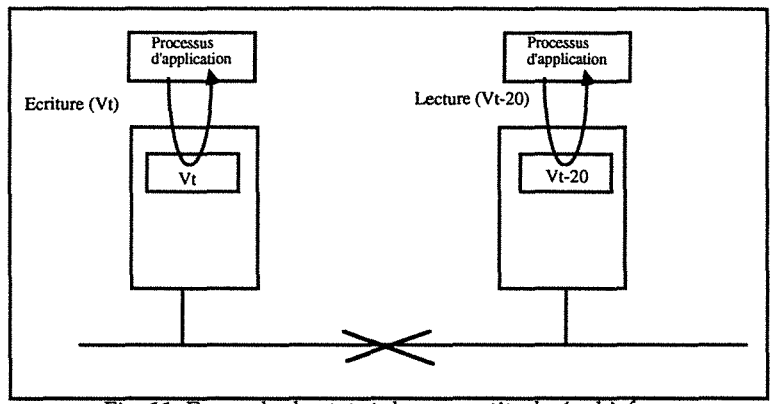


Fig. 11: Exemple de statut de promptitude égal à faux

L'élaboration du statut de promptitude se fait au moment de la réception d'une information. Ainsi dans l'exemple ci-dessus on constate que la valeur qui devrait se trouver dans le buffer de l'entité consommatrice devrait être égal à la valeur qui a été produite à l'instant t.
On peut constater que lorsque le statut de promptitude est égal à faux, l'entité consommatrice ne connaît pas, à l'instant t, la valeur du statut de rafraîchissement, puisque le statut de rafraîchissement est transmis en même temps que la valeur de la variable qui, dans ce cas, n'a pas été reçue.

Notons que si l'on différentie dans la transmission, les étapes d'émission et de réception, alors on peut représenter les cas où les statuts d'émission et/ou de réception sont à faux de la manière suivante:

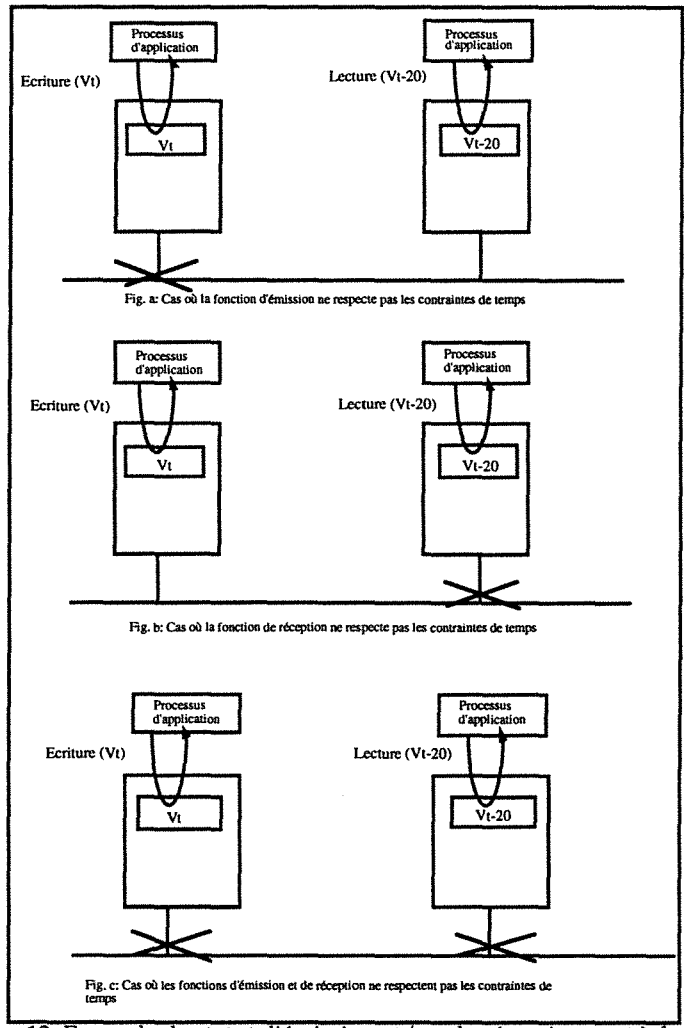


Fig. 12: Exemple de statut d'émission et/ou de réception sont à faux

Dans le réseau de terrain FIP, le statut d'émission et le statut de réception sont représentés par le statut de promptitude. Ainsi, la norme FIP [UTE 90a], ne permet pas de connaître les causes exactes du non respect des contraintes de transmission, c'est-à-dire si cela provient du non respect des contraintes d'émission ou du non respect des contraintes de réception.

4.1.3 Remarques

Si l'on combine les statuts de rafraîchissement et de promptitude, alors quatre cas peuvent se présenter:

1°) le statut de rafraîchissement est égal à vrai et le statut de promptitude est égal à vrai,
2°) le statut de rafraîchissement est égal à faux et le statut de promptitude est égal à vrai,
3°) le statut de rafraîchissement est égal à vrai et le statut de promptitude est égal à faux,
4°) le statut de rafraîchissement est égal à faux et le statut de promptitude est égal à faux.

Dans le premier cas, l'entité consommatrice sait que la production et que la transmission de l'information se sont faites en respectant les délais fixés.
Dans le second cas, l'entité consommatrice sait que la transmission s'est faite dans les délais et que la production n'a pas respecté les délais alloués.
Dans le troisième cas et dans le quatrième cas, l'entité consommatrice sait que la transmission ne s'est pas faite dans les délais, mais ne sait rien concernant la production de l'information puisque les informations concernant la production n'ont pas été transmises.

A travers cette étude, on peut déjà remarquer qu'une des faiblesses du réseau de terrain FIP provient du fait que le réseau FIP gère uniquement des statuts booléens pour la production et pour la transmission. Ainsi le réseau FIP ne permet pas de savoir, par exemple, quelle est la validité d'une variable donnée avant que cette dernière soit transmise par l'intermédiaire du médium de communication vers les entités consommatrices.

4.2 Cas du statut de fraîcheur-de-consommation

Le statut de fraîcheur-de-consommation est égal à faux lorsque la fonction de consommation ne fonctionne pas correctement.
Ainsi, dans le cas où le processus d'application de l'entité consommatrice consomme une nouvelle valeur toutes les 10 unités de temps, si la valeur contenue dans l'entité de consommation date de l'instant t et si la valeur contenue dans le processus d'application du consommateur date de l'instant t-20, alors le statut de fraîcheur-de-consommation est égal à faux.

Cet exemple peut se schématiser de la manière suivante:

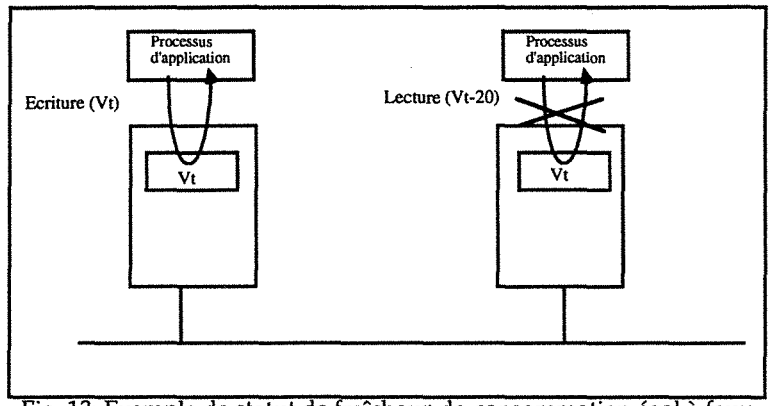


Fig. 13: Exemple de statut de fraîcheur-de-consommation égal à faux

L'élaboration du statut de fraîcheur-de-consommation se fait donc au moment de la consommation d'une information par le processus d'application.
Lorsque le statut de fraîcheur-de-consommation est égal à faux, le processus d'application de l'entité consommatrice ne peut pas savoir, à l'instant t, si les contraintes de production et de transmission ont été respectées, puisque le statut de rafraîchissement et le statut de promptitude sont transmis au processus d'application de l'entité consommatrice en même temps que la valeur de la variable qui, dans ce cas, n'a pas été transmise.

4.3 Analyse des statuts temporels dans FULLFIP

Nous allons maintenant considérer un cas pratique, en étudiant les mécanismes de promptitude et de rafraîchissement tels qu'ils sont implantés dans le composant FULLFIP [CEG 92].

FULLFIP est un co-processeur micro-programmé développé par Cegelec. C'est actuellement le plus complet des composants FIP. Il couvre la partie des services de couche physique indépendant du support de transmission, ainsi que la presque totalité des services de la couche liaison de données et de la couche application décrits dans les normes FIP [UTE 90a] et [UTE 90b].

FULLFIP offre trois interfaces:

- une interface bus sur laquelle se raccorde les outils de ligne,
- une interface système,
- une interface mémoire privée.

FULLFIP est vu comme un ensemble de registres. La mémoire privée du composant contient la base de données de la station, c'est-à-dire que l'on y retrouve l'ensemble des variables produites, des variables consommées, des tables d'arbitrage de bus ainsi que le micro-code FIPCODE qui pilote le composant.

FULLFIP est capable de gérer jusqu'à 4095 variables de 128 octets et il permet de mettre en oeuvre la fonction station ainsi que la fonction arbitrage de bus. L'outil de ligne, qui se situe entre le composant FULLFIP et le médium de communication, remonte vers la logique supérieure des erreurs d'hypocourant, d'hypercourant, de détection de bavardage, etc...

L'environnement de FULLFIP peut se schématiser de la manière suivante:

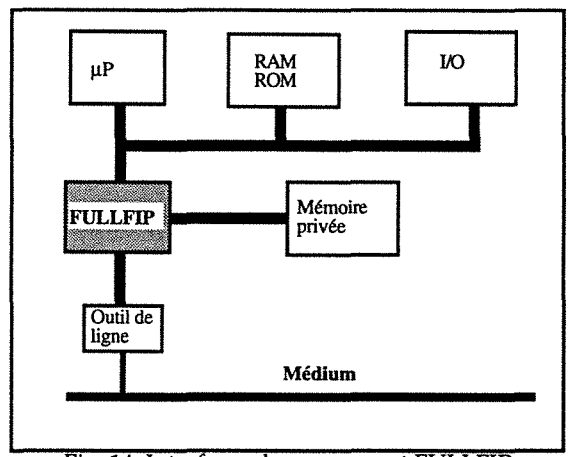


Fig. 14: Interfaces du composant FULLFIP

La version actuelle du FIPCODE [CEG 92] n'intègre que les caractères asynchrones des statuts de promptitude et de rafraîchissement.

Le statut de rafraîchissement asynchrone est égal à vrai, si l'intervalle de temps entre le moment où la variable a été produite et le moment où la variable a été transmise est inférieur à la période de production. La période de production correspond à l'intervalle de temps maximum entre le moment où la variable a été produite et le moment où cette dernière a été transmise. Le statut de rafraîchissement est élaboré au moment de l'émission de la variable sur le réseau.

Le statut de Rafraîchissement Asynchrone, noté AR, est décrit de la manière suivante dans [CEG 92]:

Si (Instant_Transmission - Instant_Production) < Période_Production
alors AR:=1 (AR est égal à vrai)
sinon AR:=0 (AR est égal à faux).

Le statut de promptitude asynchrone est égal à vrai, si l'intervalle de temps entre le moment où la variable a été reçue et le moment où la variable a été consommée est inférieur à la période de consommation. La période de consommation correspond à l'intervalle de temps maximum entre le moment où la variable a été reçue et le moment où cette dernière a été consommée. Le

statut de promptitude est élaboré au moment de la consommation de la variable par le processus d'application.

Le statut de Promptitude Asynchrone, noté AP, est décrit de la manière suivante dans [CEG 92]:

Si (Instant_Consommation - Instant_Réception) < Période_Consommation
alors AP:=1 (AP est égal à vrai)
sinon AP:=0 (AP est égal à faux).

La période de production ainsi que la période de consommation sont fixées dans la structure de données associée à la variable.

Ainsi, si on reprend notre schéma précédent, le statut de promptitude et le statut de rafraîchissement asynchrone introduits dans le FIPCODE peuvent se représenter de la manière suivante:

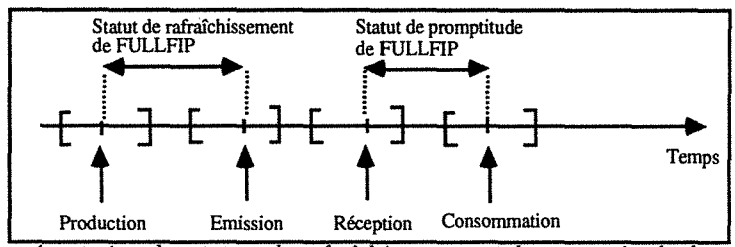


Fig. 15: Représentation des statuts de rafraîchissement et de promptitude dans FULLFIP

Si le statut de rafraîchissement asynchrone est égal à faux, alors on ne sait pas si ce sont les contraintes temporelles de production ou si ce sont les contraintes temporelles d'émission chez l'entité productrice qui n'ont pas été respectées. Ainsi, si une variable est produite dans les délais, mais si cette dernière n'est pas transmise dans les délais de transmission, alors le statut de rafraîchissement sera égal à faux.
Le statut de rafraîchissement asynchrone du FIPCODE, lorsqu'il est égal vrai, permet uniquement de savoir si l'intervalle de temps entre le moment où une variable est produite (c'est-à-dire mise dans un buffer) et le moment où cette dernière est émise sur le médium de communication a été respecté.

De même, si le statut de promptitude est égal à faux, alors on ne sait pas si ce sont les contraintes temporelles de réception ou si ce sont les contraintes temporelles liées à la consommation chez l'entité consommatrice qui n'ont pas été respectées. Ainsi, si une variable est reçue en respectant les délais, mais si elle n'est pas consommée dans les délais, alors le statut de promptitude qui est élaboré par l'entité consommatrice sera égal à faux.
Le statut de promptitude asynchrone du FIPCODE permet uniquement de savoir, si l'intervalle de temps entre le moment où la valeur d'une variable est

reçue dans un buffer et le moment où cette dernière est consommée a été respecté.

Les caractères de promptitude et de rafraîchissement implantés dans le FIPCODE ne correspondent donc pas exactement à la définition de la promptitude et du rafraîchissement telle qu'elle est décrite dans la norme FIP.

Les caractères de rafraîchissement et de promptitude sont élaborés à partir des trames FIP circulant sur le médium de communication. Les deux principaux types de trames circulant sur le médium de communication sont:

- la trame d'indication, notée ID_DAT, qui possède comme paramètre: le nom de la variable qu'une station doit produire ou consommer à un instant donné et
- la trame de réponse, notée RP_DAT, qui possède comme paramètre: le nom de la variable, la valeur de la variable ainsi que le statut de rafraîchissement.

La trame d'indication est envoyée sur le médium par l'arbitre de bus à toutes les stations connectées. Au même instant, la station productrice ainsi que les stations consommatrices se reconnaissent comme des stations productrice ou consommatrice d'une certaine variable.
Ensuite, la station productrice produit son information sur le réseau à destination de toutes les stations consommatrices par l'intermédiaire de la trame RP_DAT.
Le schéma suivant résume les mécanismes mis en oeuvre par le réseau de terrain FIP [UTE 90b]:

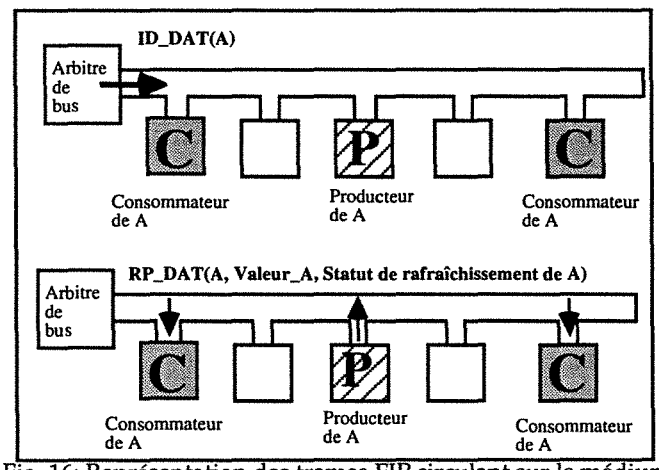


Fig. 16: Représentation des trames FIP circulant sur le médium

Notons que dans FULLFIP, le temps séparant une trame ID_DAT d'une trame RP_DAT varie de 4 µs à 28 µs lorsque le débit du réseau est de 2,5 Mbit/s. Par

contre, lorsque le débit du réseau est de 1 Mbit/s, alors le temps séparant une trame ID_DAT d'une trame RP_DAT varie de 10 µs à 70 µs.

Le FIPCODE élabore le statut de rafraîchissement en testant si la différence de temps entre le moment où la valeur de la variable est écrite dans le buffer du producteur et l'instant de l'envoi de la trame RP_DAT est inférieure à une certaine période de production fixée initialement ; dans ce cas le statut de rafraîchissement est égal à vrai. Ce statut de rafraîchissement est alors émis comme paramètre de la trame de réponse RP_DAT vers tous les différents consommateurs de cette dernière valeur.

Le FIPCODE élabore le statut de promptitude en testant si la différence de temps entre le moment où la valeur de la variable est reçue dans le buffer du consommateur par l'intermédiaire de la trame RP_DAT et l'instant de la consommation de la valeur de cette variable est inférieure à une certaine période de consommation fixée initialement ; dans ce cas le statut de promptitude est égal à vrai. Lorsque le consommateur consomme la variable, il possède deux types d'information associés à la valeur de la variable qui permettent de qualifier cette dernière: le statut de rafraîchissement et le statut de promptitude élaborés par le FIPCODE.

5. Liens entre les mécanismes temporels du réseau FIP et la machine d'états générale

Montrons maintenant que les caractères asynchrones, synchrones et ponctuels introduits dans le réseau de terrain FIP peuvent s'obtenir à partir de la machine d'états générale décrite dans le chapitre 3.

5.1 Caractère asynchrone

Le caractère asynchrone peut se déduire de la machine d'états générale:

- si on fusionne les états E-Faux, Attente, Expiration1 et Expiration2 de la machine d'état générale en un seul état, on obtient alors un état qui correspond à l'état *Statut asynchrone = FAUX,*
- l'état E-Vrai de la machine d'états générale correspond alors à l'état *Statut asynchrone = VRAI.*

Si on utilise le formalisme des réseaux de Nutt, la machine d'états présentant le caractère asynchrone, qui se déduit de la machine d'états générale, est alors la suivante:

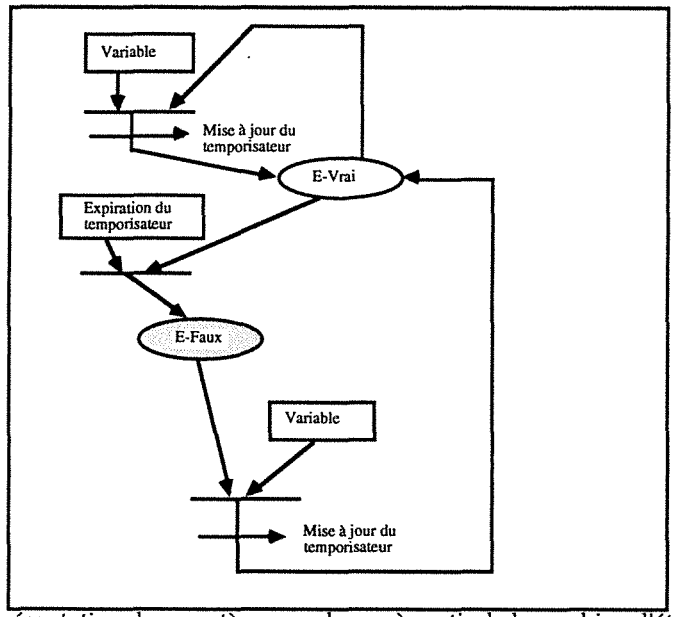


Fig. 17: Représentation du caractère asynchrone à partir de la machine d'états générale

5.2 Caractère synchrone

Le caractère synchrone peut également se déduire de la machine d'états générale:

- si l'on fusionne les états E-Faux, Expiration1 et Expiration2 de la machine d'états générale, on obtient alors l'état *Statut synchrone = FAUX,*
- l'état E-Vrai correspond à l'état *Statut synchrone = VRAI,*
- l'état Attente permet de tester si le temporisateur est armé. Si le temporisateur est armé et si on n'a pas encore reçu de "variable", alors on est dans l'état Attente. Par contre, dès que le temporisateur est échu, on passe de l'état Attente vers l'état E-Faux.

Si on utilise le formalisme des réseaux de Nutt, la machine d'états présentant le caractère synchrone, qui se déduit de la machine d'états générale, est alors la suivante:

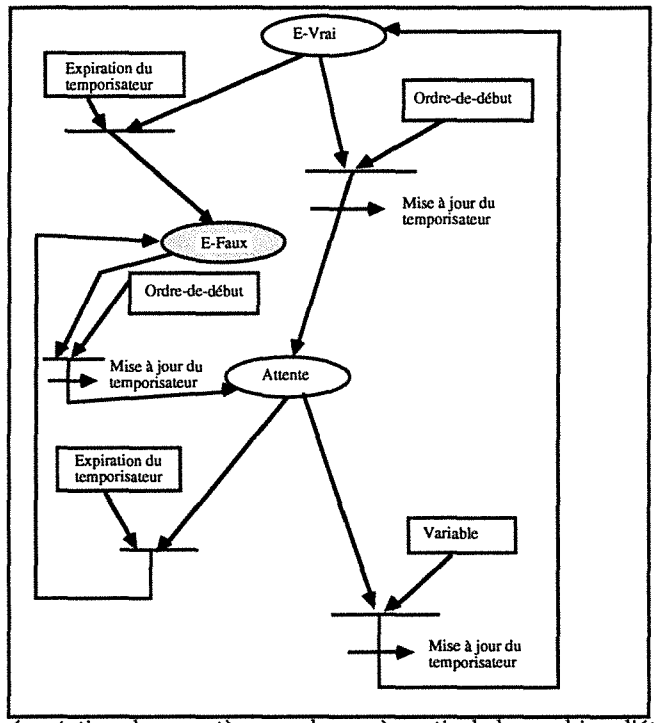


Fig. 18: Représentation du caractère synchrone à partir de la machine d'états générale

5.3 Caractère ponctuel

Le caractère ponctuel peut se déduire de la machine d'états générale:

- si l'on fusionne les états E-Faux et Expiration2 de la machine d'états générale, on obtient alors l'état *Statut ponctuel = FAUX,*
- si l'on fusionne les états E-Vrai et Expiration1 de la machine d'états générale, on obtient alors l'état *Statut ponctuel = VRAI,*
- l'état Attente permet de tester si le temporisateur est armé. Si le temporisateur est armé et si on n'a pas encore reçu de "variable", alors on est dans l'état Attente. Par contre, dès que le temporisateur est échu, on passe de l'état Attente vers l'état E-Faux.

Si on utilise le formalisme des réseaux de Nutt, la machine d'états présentant le caractère ponctuel, qui se déduit de la machine d'états générale, est la suivante:

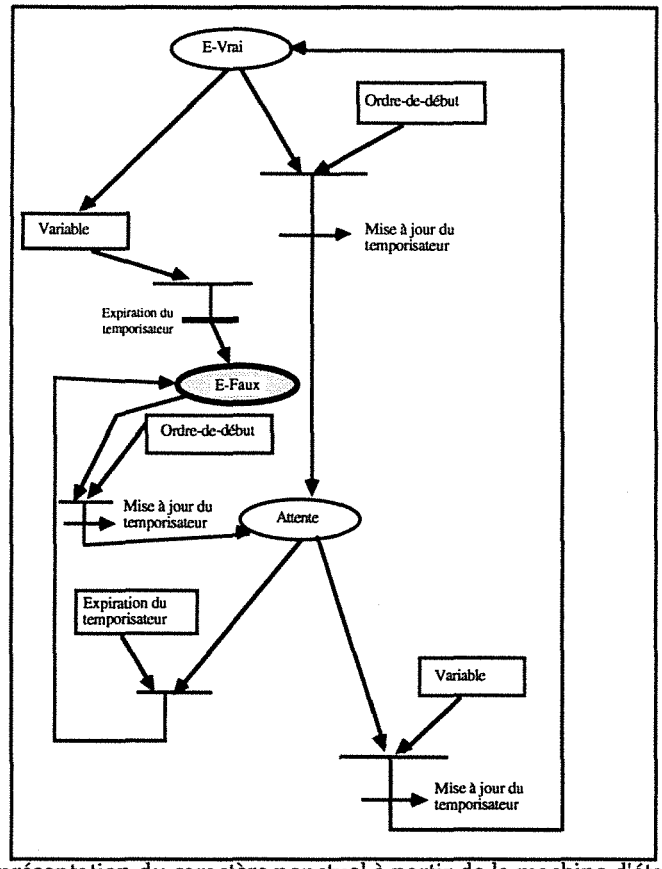


Fig. 19: Représentation du caractère ponctuel à partir de la machine d'états générale

On constate donc qu'en restreignant les fonctionnalités de notre machine d'états générale, on retrouve les caractères asynchrone, synchrone et ponctuel décrits dans le protocole de communication FIP.

6. Besoins temporels concernant les listes de variables

Dans cette partie, nous allons nous intéresser à l'élaboration des statuts temporels lors de la transmission d'une liste de variables entre plusieurs stations.
Les listes de variables vont permettre, par exemple, à un capteur qui consomme trois valeurs différentes, de faire une corrélation entre les trois valeurs reçues et de savoir ainsi si les trois valeurs consommées ont bien été produites et transmises dans un certain intervalle de temps donné. On élaborera un statut global de promptitude, de rafraîchissement et de fraîcheur-de-consommation en faisant un ET logique de l'ensemble des différentes statuts des variables composant la liste.
Les listes de variables vont également permettre de savoir, par exemple, si un ensemble de consommateurs ont bien reçus à un instant donné, les mêmes valeurs pour l'ensemble des variables composant la liste: on parlera alors de statut de cohérence spatiale.

Si l'information produite par plusieurs producteurs doit être collectée à un instant précis et si cette même information doit être consommée à un instant donné (après émission de cette information par diffusion) par un ensemble de consommateurs, il faut alors compléter les précédents mécanismes temporels que nous avons introduit pour répondre à ce besoin. Pour cela, la norme FIP introduit les concepts de buffer privé et de buffer public [UTE 90a].

6.1 Notion de buffer privé/public

Pour chaque variable, on défini chez les entités productrices ainsi que chez les entités consommatrices, un buffer privé et un buffer public.
L'accès au buffer privé en lecture ou en écriture ne peut se faire que par l'intermédiaire du processus d'application.
L'accès au buffer public en lecture ou en écriture ne peut se faire que par l'intermédiaire du médium de communication. Le processus d'application ne pourra donc jamais lire ou écrire une valeur dans un buffer public.

L'ordre de transfert d'un buffer (privé ou public) vers l'autre est nommé TOP [UTE 90a]. L'ordre TOP est composé du top de photo et du top de positionnement. Le top de photo intervient lorsque l'entité productrice reçoit l'ordre de faire une copie de la valeur de la variable se trouvant dans son buffer privé vers son buffer public. Le top de positionnement intervient lorsque l'entité consommatrice reçoit l'ordre de faire une copie de la valeur de la variable se trouvant dans son buffer public vers son buffer privé.
Les mécanismes de resynchronisation offerts par la norme FIP vont donc permettre, grâce à la fonction d'échantillonnage/blocage des valeurs produites et consommées, à des processus d'application asynchrones de fonctionner avec des applications réparties synchrones [UTE 90a].

Pour simplifier notre étude, nous utiliserons le terme de buffer d'émission (introduit dans le chapitre 1 dans le cadre de l'étude du modèle de communication producteurs/consommateurs), pour parler du buffer privé chez l'entité productrice et du buffer public chez l'entité consommatrice.
De même, nous utilisons le terme de buffer de réception (introduit dans le chapitre 1 dans le cadre de l'étude du modèle de communication producteurs/consommateurs) pour parler du buffer public chez l'entité productrice et du buffer privé chez l'entité consommatrice.

Les mécanismes de resynchronisation associés aux buffers privés et aux buffers publics peuvent se schématiser de la manière suivante :

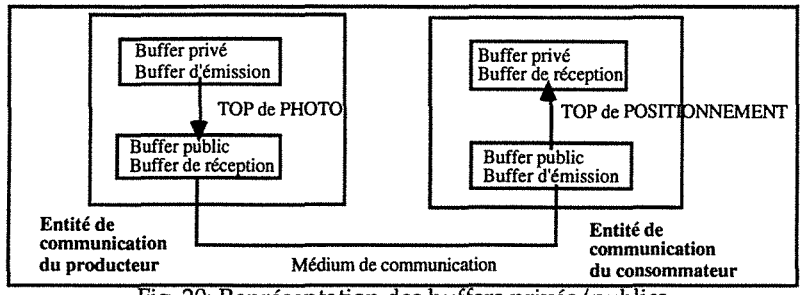


Fig. 20: Représentation des buffers privés/publics

6.2 Représentation des machines d'états temporelles pour les listes de variables

Concernant les caractères temporels des buffers privés et des buffers publics, la norme FIP [UTE 90a] introduit les notions de statut de rafraîchissement et de promptitude qui peuvent posséder les caractères asynchrones, synchrones et ponctuels.

Nous allons utiliser à nouveau le formalisme des réseaux de Nutt pour représenter les différentes machines d'états permettant d'élaborer les différents statuts public et privé de fraîcheur-de-consommation.

Le statut public de fraîcheur-de-consommation asynchrone chez l'entité consommatrice, peut se décrire à l'aide de la machine d'états suivante:

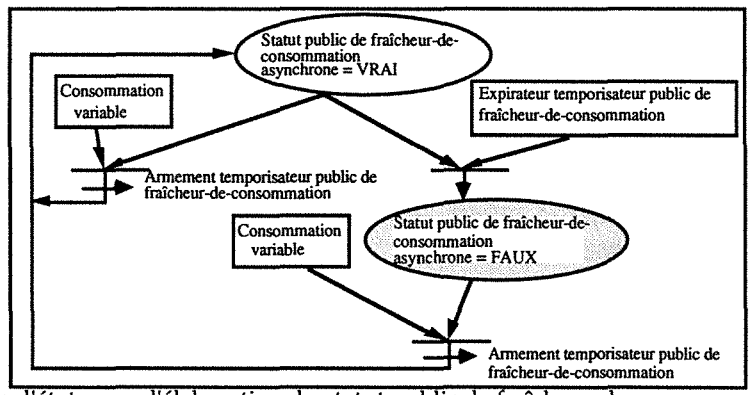


Fig. 21: Machine d'états pour l'élaboration du statut public de fraîcheur-de-consommation asynchrone

Nous n'avons pas représenté les machines d'états temporelles pour le statut public de fraîcheur-de-consommation synchrone et ponctuel, puisque les machines d'états d'émission sont identiques à celles décrites dans le paragraphe 3.3.

L'élaboration de la machine d'états pour l'élaboration du statut privé de fraîcheur-de-consommation prend en compte le statut public de fraîcheur-de-consommation. Ainsi, le statut privé de fraîcheur-de-consommation asynchrone peut se décrire à l'aide de la machine d'états suivante:

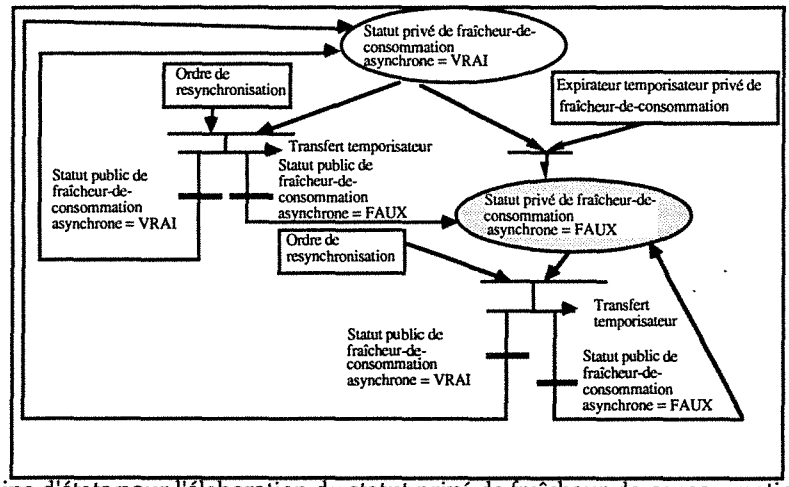


Fig. 22: Machine d'états pour l'élaboration du statut privé de fraîcheur-de-consommation asynchrone

Il est important de souligner que les machines d'états de réception, concernant l'élaboration des statuts publics de rafraîchissement et des statuts privés de promptitude et de fraîcheur-de-consommation, sont liées aux machines d'états d'émission.

Lorsqu'une machine d'états d'émission a un statut égal à faux, alors le statut de la machine d'états de réception qui lui est liée est également égal à faux. L'expiration du temporisateur de la machine d'états de réception doit toujours intervenir avant ou en même temps que l'expiration du temporisateur de la machine d'états d'émission. Par contre, lorsque le statut de la machine d'états d'émission est égal à vrai, alors le statut de la machine d'états de réception peut être égal à vrai ou à faux suivant que l'ordre de resynchronisation a été reçu ou non.

Pour les caractères synchrones et ponctuels, les machines d'états temporelles pour l'élaboration du statut privé de fraîcheur-de-consommation peuvent se représenter de la manière suivante:

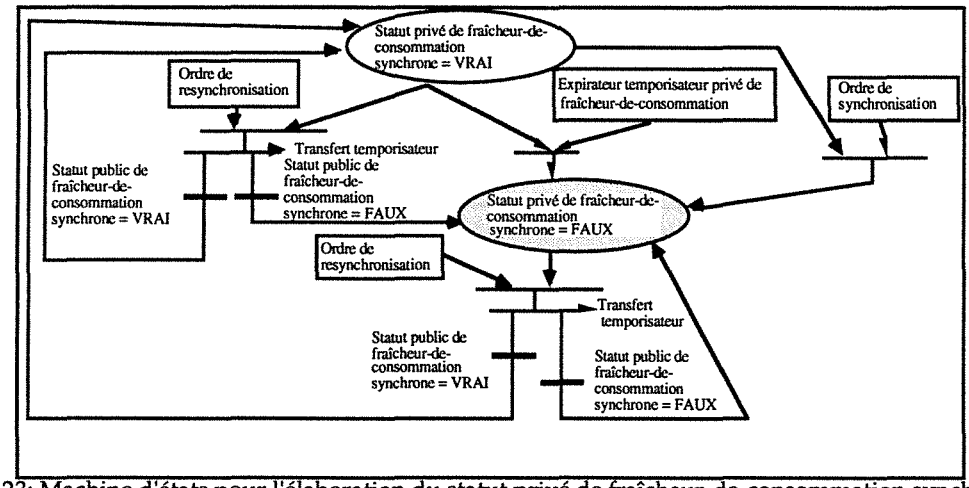


Fig. 23: Machine d'états pour l'élaboration du statut privé de fraîcheur-de-consommation synchrone

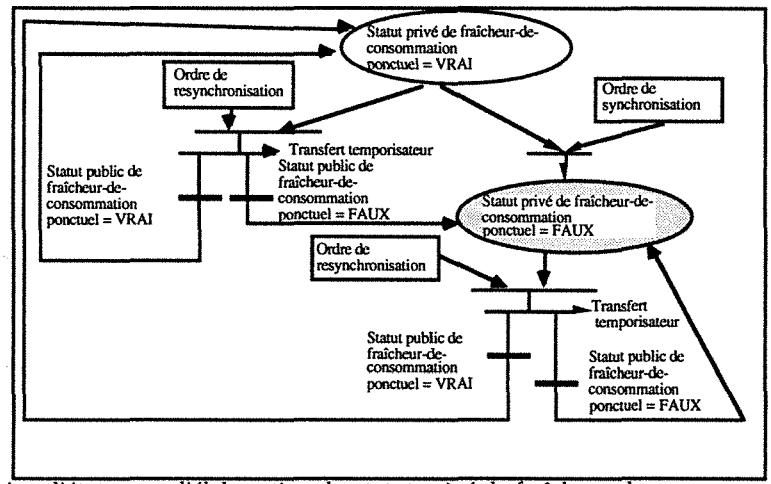


Fig. 24: Machine d'états pour l'élaboration du statut privé de fraîcheur-de-consommation ponctuel

6.3 Statut de cohérence spatiale

Dans les listes de variables, il est possible d'introduire un statut de cohérence spatiale [UTE 90a] permettant de savoir si tous les consommateurs d'une liste de variables ont bien reçu les mêmes valeurs.
Il est alors possible de combiner les statuts de cohérence spatiale avec les statuts temporels introduits précédemment, et ceci afin de savoir si en plus de l'identité des valeurs d'une liste de variables chez les différents consommateurs, si ces dernières sont valides temporellement.

Un exemple de représentation d'élaboration d'un statut de cohérence spatiale égal à vrai peut se représenter de la manière suivante:

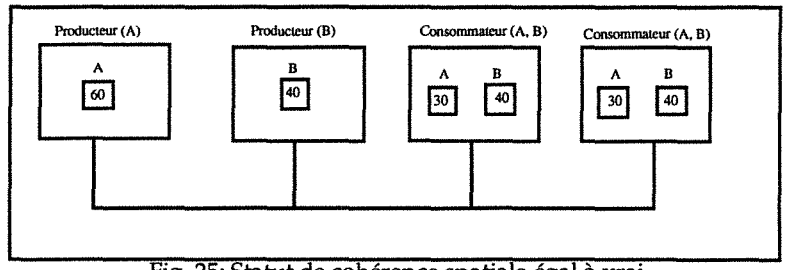


Fig. 25: Statut de cohérence spatiale égal à vrai

Les deux consommateurs de la liste de variables (A, B) ont des valeurs identiques pour chacune des variables A et B, donc le statut de cohérence spatiale est égal à vrai.
De plus, si les étapes de production, de transmission et de consommation des variables A et B ont respectées les contraintes de temps fixée, alors les statuts de rafraîchissement, de promptitude et de fraîcheur-de-consommation associés à ces deux variables seront égaux à vrai.

Toujours dans le même exemple décrit ci-dessus, on peut constater que chez le producteur, la valeur de la variable A est différente de celle qui se trouve chez les consommateurs. Donc le statut de cohérence spatiale tel qu'il est décrit dans la norme FIP [UTE 90a], ne permet pas de savoir, à un instant donné, si les différentes valeurs d'une liste de variables qui se trouve chez les consommateurs sont identiques à celles situées chez les producteurs.

Ainsi pour les listes de variables, il est possible d'utiliser de manière séparée ou combinée les mécanismes de resynchronisation, les statuts de cohérence spatiale et les statuts de cohérence temporelle.

7. Conclusion

Dans ce chapitre, nous avons montré que la machine d'états générale introduite dans le chapitre 3, permet de retrouver les mécanismes temporels de rafraîchissement et de promptitude ainsi que les caractères asynchrones, synchrones et ponctuels décrits dans le réseau de terrain FIP.

Les mécanismes temporels du réseau FIP ont été complétés, en introduisant les mécanismes temporels de fraîcheur-de-consommation qui peuvent également se déduire de la machine d'états générale.

Nous avons ensuite étudié les insuffisances des mécanismes temporels du réseau de terrain FIP ainsi que l'implantation de ces mécanismes dans le composant FULLFIP. Ceci nous a permis de montrer que l'implantation actuelle des mécanismes temporels de FIP par FULLFIP n'est pas tout-à-fait conforme à la norme FIP.

Dans une dernière partie, nous nous sommes intéressés aux mécanismes associés aux listes de variables. Nous avons introduit des machines d'états permettant de s'assurer que tous les éléments d'une liste de variables ont bien été produits et consommés en même temps et à un instant donné en respectant les contraintes de temps.

Conclusion générale

Conclusion générale

Le temps réel a fait l'objet, ces dernières années, de nombreux travaux notamment dans le cadre des réseaux de terrain. Le rôle joué par les réseaux de terrain s'est révélé très important et de nombreuses rivalités sont nées entre les divers concepteurs de réseaux, et ceci afin d'imposer leur standard dans le cadre de la normalisation internationale.

La démarche utilisée a été tout d'abord basée sur l'étude et sur la définition d'un ensemble de modèles de communication. Ensuite, nous avons présenté une approche permettant de spécifier, par l'intermédiaire des mécanismes de fenêtres temporelles, différentes contraintes de temps exprimées en terme de date au plus tôt et de date au plus tard.
Ainsi, nous avons présenté un ensemble de relations entre les fenêtres temporelles de production, d'émission, de réception et de consommation, ainsi qu'entre les fenêtres temporelles de validité des variables produites et consommées.

Ceci nous a permis de construire une machine d'états générale permettant de nous informer sur le respect des contraintes de temps durant les différentes étapes de la communication.
Nous avons ensuite montré que notre machine d'états générale pouvait s'appliquer à plusieurs modèles de communication tels que les modèles client(s)/serveur(s) et producteur(s)/consommateur(s). De plus, la machine d'états générale introduite nous a permis de retrouver et de compléter les mécanismes temporels du réseau de terrain FIP, d'où une première utilisation possible dans le monde industriel.

L'apport original de notre contribution se situe dans la formalisation des aspects temporels. Au niveau des résultats obtenus, notre travail s'est concrétisé par l'élaboration d'une machine d'états générale qui a été validée de manière formelle et par une implantation de cette machine d'états qui permet de montrer son utilisation dans le cadre d'applications réelles.

Les difficultés pour mener à bien notre projet ont été:

- la nécessité d'identifier les nombreux formalismes existant pour exprimer des contraintes de temps,
- la nécessité d'effectuer une large recherche bibliographique sur les différents travaux déjà existants dans ce domaine,
- la nécessité de présenter des mécanismes pouvant s'appliquer à un problème industriel.

Les prolongements de ce travail peuvent s'effectuer dans les directions suivantes:

- par l'étude de méthodes permettant de générer automatiquement des contraintes de temps à partir des besoins des utilisateurs et de vérifier la cohérence de ces derniers à partir des règles introduites;

- par la recherche de méthodes permettant d'intégrer les mécanismes temporels au niveau des services et des protocoles et de vérifier ensuite leur interopérabilité avec d'autres systèmes;

- par le développement des mécanismes temporels introduits, afin de pouvoir les appliquer dans le cas où les contraintes temporelles de date au plus tôt et de date au plus tard se situent sur des sites différents;

- et d'un point de vue plus pratique, par la mesure des performances obtenues par différents réseaux de communication intégrant les mécanismes temporels.

Bibliographie

Bibliographie

[ALA 92] M. Alabau, T. Dechaize, "Ordonnancement temps réel par échéance", TSI, vol 11, n°3, 1992, pp. 59-123.

[ALL 83] J.F. Allen, "Maintaining knowledge about temporal intervals", Communication of the ACM, vol 26, n°11, 1983, pp. 832-843.

[ALL 84] J.F. Allen, "Towards a general theory of action and time", Artificial Intelligence, n°23, 1984, pp. 123-154.

[ARN 89] A. Arnold, "Mec: a system for constructing and analysing transition system", Proceedings of the Workshop on Automatic Verification Methods for Finite State Systems, Grenoble, 1989.

[AUD 91] E. Audureau, P. Enjalbert, L. Farinas del Cerro, "Logique temporelle: sémantique et validation de programmes parallèles", Edition Masson.

[BEL 88] F.R. Belmans, O.D. Kaitouni, "Spécification et validation d'un protocole de communication adapté au temps réel", IRISA, Publication interne n°394, Rennes, Février 1988.

[BER 92] N. Bergé, M. Samaan, "Application temps réel sur une architecture distribuée: premiers éléments de réflexion", rapport interne EDF n°HM-63/9659, 1992.

[BOU 85] G. Boudol, "Parallélisme, communication et synchronisation", édité par J.-P. Verjus et G. Roucairol, édition du CNRS, 1985.

[BOU 89] G. Boudol, V. Roy, R. Simone, D. Vergamini, "Process calculi, from theory to practice: verification tools", Rapport de recherche n°1098, INRIA, October 1989.

[BRE 93] H. Brenier, "CIM et temps réel", édition DUNOD, 1993.

[BRO 87] P.F. Brown, C.R. McLean, "The architecture of the NBS factory automation research tested", Congrès IFAC, Münich, RFA, 1987, pp. 153-179.

[BRO 89] M. Broy, "Functional specification of time sensitive communication system", Lecture Notes in Computer Science n°430, Stepwise refinement of distributed systems, REX Workshop, Mook, The Netherlands, May-June 1989.

[CAR 93a] C. Cardeira, Z. Mammeri, J.P. Thomesse, " Scheduling tasks and traffic in fieldbus based real-time systems", Real Time Computing, W.A. Halang and A.D. Stoyenko - Springer Verlag, 1993.

[CAR 93b] C. Cardeira, Z. Mammeri, J.P. Thomesse, "Ordonnancement dans les systèmes temps réel et répartis", Rapport interne n°93-R-097, CRIN, Université de Nancy, 1993.

[CHE 90] H. Chetto "L'ordonnancement dans les systèmes de contrôle temps-réel à contraintes strictes", Thèse d'état, Université de Nantes, Décembre 1990.

[CRU 89] P. Crubillé, "Réalisation de l'outil Mec, spécification fonctionnelle et architecture", Thèse de doctorat d'informatique, Université de Bordeaux, 17 novembre 1989.

[DAK 90] Y. Dakroury, "Spécification et validation d'un protocole de messagerie multi-serveurs pour l'environnement MMS", Thèse de doctorat, Université de Nantes, 21 novembre 1990.

[DAS 85] B. Dasarathy, "Timing constraints of real-time systems: constructs for expressing them, methods of validating them", IEEE Transactions on Software Engineering, vol SE-11, n°1, January 1985, pp. 80-86.

[DEC 93] J.D. Decotignie, P. Raja, "Fulfilling temporal constraints in fieldbus", IECON'93, vol 1, 15-19 November 1993, Hawaii, USA, pp. 519-524.

[DEL 89] J.L. Delcuvellerie, "Ingénierie des systèmes d'automatisation de production", Thèse de doctorat en informatique, Institut National Polytechnique de Lorraine, Nancy, Mars 1989.

[DER 82] D.V. McDermott, "A temporal logic for reasoning about processes and plans", Cognitive Science, vol 6, 1982, pp. 101-155.

[DIA 92] M. Diaz, "Conception formelle des protocoles et des services dans les systèmes distribués", Génie logiciel&systèmes experts, n°26, Mars 1992, pp. 57-70.

[DID 93] M. Didic, F. Neuscheler, L. Bogdanowicz, "McCIM: execution of CIMOSA Models", Proceedings of the Ninth CIM-Europe, Amsterdam, 12-14 May 1993.

[ELL 90] J.P. Elloy, "Les contraintes du temps réel dans les systèmes industriels répartis", Journée d'études: Les exigences du temps réel sur les réseaux locaux, SEE, Grenoble, 6 juin 1990.

[ETT 93] M. Ettl, U. Klehmet, "A performance comparison between the fieldbus protocol standards PROFIBUS and FIP", IFIP International Conference on Decentralized and Distributed Systems, Palma de Mallorca, Spain, 13-17 September 1993, pp. 245-258.

[GAL 84] D. Galara, J.P. Thomesse, "Groupe de réflexion FIP, proposition d'un système de transmission série multiplexée pour les échanges d'informations entre des capteurs, des actionneurs et des automates réflexes", Ministère de l'industrie et de la recherche, 1984.

[GOD 93] P. Godefroid, G.J. Holzmann, "On the verification of temporal properties", 13th IFIP Symposium on Protocol Specification, Testing and Verification, Liège, Belgium, 25-28 May 1993, pp. B3.1-B3.16.

[GOT 93] Groupe de réflexion GOTHA, "Les problèmes d'ordonnancement", Recherche opérationnelle, vol 27, n°1, 1993, pp. 77-150.

[GRE 85] GREPA (Groupe Equipement de Production Automatisée), "Le Grafcet", Editions Cepadues, 1985.

[HAT 91] J.P. Haton, N. Bouzid, F. Chapillot, M.C. Haton, B. Lâasri, H. Lâasri, P. Marquis, T. Mondot, A. Napoli, "Le raisonnement en intelligence artificielle", Edition InterEditions, 1991.

[HE 93] J. He, "Modélisation de la synchronisation des horloges dans les systèmes répartis", Thèse de doctorat en informatique, Institut National Polytechnique de Lorraine, Nancy, 7 juillet 1993.

[JON 86] A.T. Jones, C.R. McLean, "A proposed hierarchical control model for automated manufacturing systems", Journal of manufacturing system, vol 5, n°1, 1986.

[JUA 92] G. Juanole, Y. Atamna, N. Bergé, "Modelling time critical communication networks with stochastic timed petri nets", International workshop on real-time programming, WRTP'92, Bruges, Belgium, 23-26 June 1992, pp. 143-148.

[KAT 93] M. Katz, A. Funke, G. Biwer, Y. Li, T. Sebastiani, B. Rieger, "PROFIBUS: the fieldbus for industrial automation", Edited by K. Bender, Carl Hanser Verlag, Prentice Hall, 1993.

[KIR 91] T.G. Kirner, "Real-time systems specification: a process and a tool", ACM Sigsoft, Software engineering notes, vol 16, n°1, 1991, pp. 62-74.

[KOP 87] H. Kopetz, W. Ochsenreiter, "Clock synchronization in distributed real-time systems", IEEE Transaction on computers, vol C-36, n°8, August 1987, pp. 933-940.

[KOP 90] H. Kopetz, K.H. Kim, "Temporal uncertainties in interactions among real-time objects", 9th Symposium on reliable distributed systems, Huntsville, Alabama, USA, 9-11 october 1990, pp. 165-174.

[KOP 92] H. Kopetz, "Sparse time versus dense time in distributed real-time systems", 12th International conference on distributed computing systems, Yokohama, Japan, 9-12 June, 1992, pp. 460-467.

[KOT 93] I.L. Kotsiopoulos, "Theorical aspects of CIM-OSA modelling", Proceedings of the Ninth CIM-Europe, Amsterdam, 12-14 May 1993.

[LAD 86] P. Ladkin, "Time representation: a taxonomy of interval relations", Proceedings of AAAI, Science, 1986, pp. 360-366.

[LAM 78] L. Lamport, "Time, clocks and the ordering of events in a distributed system", Communication of the ACM, vol 21, n°7, July 1978, pp. 558-565.

[LAM 83] L. Lamport, "What good is temporal logic?", Information Processing 83, Elsevier Science Publishers, IFIP, 1983, pp. 657-668.

[LAV 88] R. Lavie, "The CIM Integrated Data Processing Environment in the European open systems architecture CIM-OSA", MAP/TOP COS Enterprise Networking Event '88 International, Baltimore, USA, 6-8 June 1988.

[LEC 93] J. Lecuivre, F. Simonot, J.P. Thomesse, "CIM applications modelling and 'traceability': from requirements to a real time distributed solution", IEEE International Conference on System, Man and Cybernetics, Le Touquet, 17-20 October 1993, pp. 548-554.

[LEL 86] G. LeLann, "Distributed real-time processing", Computer systems for process control, Plenum Press, 1986.

[LEL 91] G. LeLann, G. Bres, "Reliable atomic broadcast in distributed systems with omission faults", ACM Operating systems review, vol 25, n°2, April 1991, pp. 80-86.

[LIE 92] C.C. Lien, C.C. Yang, "Reduction of useless services with timing constraints", Proceedings of the 3rd IEEE workshop on future trends of distributed computing system, Taipei, 14-16 April 1992, pp. 226-231.

[LOR 90] P. Lorenz, "Contribution à l'étude de la gestion de réseaux et du temps critique dans le contexte du bus de terrain FIP", Rapport de DEA, CRIN, Université de Nancy, Septembre 1990.

[LOR 91] P. Lorenz, Z. Mammeri, J.P. Thomesse, "Time-critical network management for FIP fieldbus in industrial applications", Workshop "CIM in the Process Industry", Commission of the European Community, Athens, Greece, 14-15 March 1991.

[LOR 93a] P. Lorenz, Z. Mammeri, J.P. Thomesse, "Time management in communication for real time applications", Proceedings of the Ninth CIM-Europe Annual Conference, Rai Amsterdam, The Netherlands, 12-14 May 1993, pp. 161-169.

[LOR 93b] P. Lorenz, F. Beaudouin, M. Castello, "Intelligent Transmitters: Needs for Interoperability and Interchangeability", Sensor 93, 6th International Exhibition and Congress for Sensor and System Technology, Nürnberg, Germany, 11-14 October 1993, pp. 75-85.

[LOR 94a] P. Lorenz, Z. Mammeri, J.P. Thomesse, "A State-Machine for Temporal Qualification of Time-Critical Communication", 26th IEEE Southeastern Symposium on System Theory, Ohio, USA, 20-22 March 1994, pp. 654-658.

[LOR 94b] P. Lorenz, F. Beaudouin, "Temporal Communication Mechanisms for Intelligent Sensors", IMACS - IEEE/SMC International Symposium on Signal Processing, Robotics and Neural Networks, SPRANN'94, Lille, France, 25-27 April 1994, pp. 359-362.

[LOR 94c] P. Lorenz, Z. Mammeri, "Temporal Mechanisms in Communication Models applied to Companion Standards", SICICA'94, 2nd IFAC Symposium on Intelligent Components and Instruments for Control Applications, Budapest, Hungary, 8-10 June 1994, pp. 92-97.

[MAL 92] N. Malcolm, W. Zhao, "Advances in hard real-time communication with local area networks", 17th IEEE Conference on local computer network, Minneapolis, 13-16 September 1992, pp. 548-557.

[MAM 94] Z. Mammeri, P. Lorenz, "Integration of temporal mechanisms in communication protocols for time-critical distributed systems", 12th IFAC Workshop on Distributed Computer Control Systems, DCCS'94, 28-30 September 1994, Toledo, Spain.

[MAR 88] D.C. Marinescu, "A splitting algorithm for communication with real-time delivery constraints", Twenty-Sixth Annual Allerton Conference on Communication, Control and Computing, vol 2, 1988, pp. 955-64.

[MEI 91] I. Meiri, "Combining qualitative and quantitative constraints in temporal reasoning", AAAI, 1991, pp. 260-267.

[MER 76] P.M. Merlin, "A methodology for design and implementation of protocols", IEEE Transactions on Communications, n°6, June 1976, pp. 614-621.

[MIL 83] R. Milner, "Calculi for Synchrony and Asynchrony", Theoretical Computer Science 25, 1983, pp. 267-310.

[MOT 92] L. Motus, "Time concepts in real-time software", IFAC/IFIP, International workshop on real-time programming, WRTP'92, Bruges, Belgium, 24-26 June 1992.

[NAI 92] Y. Naik, "A temporal approach to requirements specification of real-time systems", Formal techniques in real-time and fault tolerant systems, 2nd international Symposium, The Netherlands, 8-10 January 1992, pp. 341-361.

[NIC 91] X. Nicollin, J. Sifakis, "An overview and synthesis on timed process algebras", 3rd international workshop CAV'91, Danmark, 1-4 July 1991, pp. 376-398.

[NIV 82] N. Nivat, A. Arnold, "Comportements de processus", Colloque AFCET "Les mathématiques de l'informatiques", 1982, pp. 35-68.

[NOR 93] J.D. Northcutt, E.M. Kuerner, "System support for time-critical applications", Computer communications, vol 16, n°10, October 1993, pp. 619-636.

[NUT 72] G. Nutt, "Evaluation nets for computer performance analysis", AFIPS FJCC, 1972, pp. 279-286.

[OST 91] J.S. Ostroff, "Temporal logic for real-time systems", Research studies press LTD, 1991.

[PER 85] G.R. Perrin, "La communication: un outil pour la spécification, la construction et la vérification de systèmes parallèles", Thèse de d'état, Université de Nancy, 1989.

[RAJ 91] R. Rajkumar, "Synchronization in real-time systems, a priority inherence approach", Kluwer academic publishers, 1991.

[RAJ 92] C.V. Raju, R. Rajkumar, F. Jahanian, "Monotoring timing constraints in distributed real-time systems", Proceeding of the real-time system symposium, Phoenix, 2-4 December, 1992.

[RAJ 93] P. Raja, G. Noubir, L. Ruiz, J. Hernandez, J.D. Decotignie, "Analysis of polling protocols for fieldbus networks", ACM Sigcomm, Computer Communication Review, vol 23-3, July 1993, pp. 69-89.

[RAM 74] C. Ramchandani, "Analysis of asynchronous concurrent systems by Petri nets", Research Report MAC-TR 120, MIT, February 1974.

[RAM 89] K. Ramamritham, J.A. Stankovic, "Time-constrained communication protocols for hard real-time systems", 6th IEEE Workshop on Real Time Operating System and Software, Pittsburg, Pensylvania, Mai 1989, pp. 61-66.

[RAY 91] M. Raynal, "La communication et le temps dans les réseaux et les sytèmes répartis", CEA-EDF INRIA Ecole d'été d'informatique, Editions Eyrolles, 1991.

[RAZ 89] R. Razouk, M. Gorlick, "A real-time interval logic for reasoning about executions of real-time programs", ACM Sigsoft' 89, Third Symposium on Software testing, Analysing and Verification, 1989, pp. 10-19.

[REU 91] C. Reutenauer, "Aspects mathématiques des réseaux de Pétri", Etudes et recherches en informatique, Edition Masson, 1991.

[ROU 88] T.J. Routt, "From TOP (3.0) to bottom: architectural close-up", Data Communication, Mai 1988.

[ROY 89] V. Roy, R. Simone, "An Autograph primer", Technical Report n°112, INRIA, October 1989.

[SAH 88] A. Sahraoui, "Timing constraints problems in real-time systems", Rapport LAAS-CNRS Technical Report n°88346, Toulouse, October 1988.

[SAH 92] A. Sahraoui, D. Delfieu, "Zaman, a simple langage for expressing timing constraints", International workshop on real-time programming, WRTP'92, Bruges, 23-26 June 1992, pp. 19-24.

[SCH 83] Swartz, Melliar-Smith, Vogt, "An interval-based temporal logic", Lecture Notes Computer Sciences, n°165, 1983, pp. 443-457.

[SCH 92] F.A. Schreiber, "Is time a real time? An overview of time ontology in informatics", NATO, Advanced study Institute on real-time computing, 5-18 October, 1992, Sint Maarten, Dutch Antilies.

[SHA 92] A.C. Shaw, "Communicating real-time state machine", IEEE Transactions on software engineering, vol 18, n°9, September 1992, pp. 805-816.

[SHE 90] D. Shepherd, M. Salmony, "Extending OSI to support synchronization required by multimedia application", Computer Communications, vol 13, n°7, September 1990.

[SHI 92] K.G. Shin, C.C. Chou, "Design and evaluation of real-time communication for fieldbus based manufacturing systems", 17th IEEE Conference on local computer network, Minneapolis, 13-16 September 1992, pp. 483-492.

[SHI 93] K.G. Shin, Q. Zheng, "Mixed time-constrained and non-time-constrained communications in local area networks", IEEE Transaction on Communications, vol 41, n°11, November 1993, pp. 1668-1676.

[SIM 89] R. Simone, D. Vergamini, "Aboard Auto", Technical report n°111, INRIA, October 1989.

[SIM 90] F. Simonot, Y.Q. Song, J.P. Thomesse, "Reliability of time critical multicast communication", IECON'90, Pacific grove , USA, November 1990, pp. 517-521.

[SIM 92a] F. Simonot, J.P. Thomesse, G. Bianchi, C. Courrier, "Architecture des systèmes", Deuxième colloque Automatique et informatique avancé dans la sidérurgie, INPL, Nancy, 14 mai 1992.

[SIM 92b] F. Simonot, C. Verlinde, "Importance d'un cadre de référence dans la mise en place d'une démarche de développement d'un système automatisé de production", Actes conférence Canadienne sur l'automatisation industrielle, Monréal, Canada, Juin 1992.

[SON 91] Y.Q. Song, P. Lorenz, F. Simonot, J.P. Thomesse, "Multipeer/Multicast Protocols for Time-critical Communication", Multipeer/Multicast Workshop for Initiating a U.S. Program to Develop the Architecture, Service Definition and Protocol Specifications, Orlando, USA, 6-8 August 1991.

[SPR 89] B. Sprunt, L. Sha, J.P. Lehoczky, "Aperiodic Task Scheduling for Hard Real-Time Systems", Journal of Real-Time Systems, n°1, June 1989, pp. 27-60.

[THO 86] J.P. Thomesse, J.L. Delcuvellerie, J.C. Derniame, F. Lepage, E. Pichon, A. Schaff, "Les Réseaux Locaux Industriels", Editions EC2, Nanterre, 1986.

[THO 89a] J.P. Thomesse, T. Laine, "The fieldbus application Services", Proceedings IECON'89, 15th Conference IEEE-IES Factory automation, Philadelphia (USA), November 1989, pp. 526-530.

[THO 89b] J.P. Thomesse, P. Noury, "Communication models. Client-server vs Producer-Distributor-Consumer", Contribution ISO TC184/SC5/WG2-TCCA, October 1989.

[THO 90a] J.P. Thomesse, "Application Architecture and Local Area Networks", Proceedings BIAS'90, Milano, Italy, November 1990.

[THO 90b] J.P. Thomesse, "Les réseaux locaux à temps critique", Colloque AFCET Le Temps Réel, Automatique, Productique et Informatique, Nantes, 10-11 octobre 1990, pp. 29-40.

[THO 91a] J.P. Thomesse, P. Lorenz, "Fieldbus - Communications models - Real time data base", Giornata di studio "Il fieldbus per la comunicazione nei sistemi di controllo di processo e di controllo automatizzato della produzione", Associazione Nazionale Industrie Elettriche e Elettroniche, Milano, Italy, 6 June 1991, pp. 5-14.

[THO 91b] J.P. Thomesse, P. Lorenz, J.P. Bardinet, P. Leterrier, T. Valentin, "Factory Intrumentation Protocol: Model, Products and Tools", Control Engineering, vol 38, n°12, September 1991, pp. 65-67.

[THO 93] J.P. Thomesse, "Le réseau FIP", Réseau et Informatique Répartie, vol 3, n°3, pp. 287-321.

[TOL 92] H. Tolba, "Contribution à l'étude du raisonnement temporel: intégration des informations qualitatives et quantitatives et propagation de contraintes", Thèse de doctorat en informatique, Université de Nancy, Décembre 1992.

[VAS 93] F. Vasques, G. Juanole, "Fieldbus MAC mechanisms for hard real time data communication support", OBS'93, Open Bus Systems, Vita Congress, Münich, Germany, 29-30 November 1993, pp. 225-232.

[VEG 93] L. Vega, F. Simonot, J.P. Thomesse, "A service oriented approach for structuring distributed control system", Proceedings of IMACS/IFAC Second International Symposium on Mathematical and Intelligent Models in System Simulation, vol 1, Brussels, Belgium, 12-16 April 1993, pp. 383-391.

[VER 89] C. Verlinde, "Contribution à l'étude des architectures de systèmes automatisés", Thèse de doctorat en informatique, Institut National Polytechnique de Lorraine, Nancy, Février 1989.

[WIL 92] N.J. Wilson, "A survey of temporal logic methods in system development", The Unified Computation Laboratory, The Institute of Mathematics and its Applications, Oxford University Press, 1992, pp. 243-258.

[XU 91] J. Xu and D.L. Parnas "On satisfying timing constraints in hard-real-time systems", Proceedings of the ACM Sigsoft'91 Conference on Software for Critical Systems, New Orleans, 4-6 December, 1991, pp. 132-146.

[ZAK 84] Zakari, "FLEXI: Langage de conception d'application de conduite de procédés industriels, contribution à l'étude de la communication entre processus coopérants", Thèse de doctorat 3ème cycle, Université de Nancy, 1984.

Documents normatifs

[ESP 92] "CIM-OSA Reference Base I and II", ESPRIT II Project 2422 AMICE Consortium, 1992.

[DIN 90] Deutsches Institut für Normung, "Profibus", Normes DIN 19 245, part 1 and part 2, 1990.

[ISO 89a] ISO 7498, Open Systems Interconnection, Basic reference model, 1989.

[ISO 89b] "Manufacturing Message Specification (MMS), Part 1: Service Definition, Part 2: Protocol Specification", ISO 9506, 1989.

[ISA 92] ISA Instrument Society of America, "User layer", Technical report, report number 389E, August 1992.

[ISO 93] ISO TR 12178, "User requirements for systems supporting time-critical communications", 1993.

[MAP 88] MAP 3.0, General Motors, 1988.

[UTE 89] FIP Gestion de Réseau, Union Technique de l'Electricité, Pr C46-605, Courbevoie, 1989.

[UTE 90a] FIP Couche Application MPS, NF C46-602, Union Technique de l'Electricité, Courbevoie, 1990.

[UTE 90b] FIP Couche Liaison de Données, NF C46-603, Union Technique de l'Electricité, Courbevoie, 1990.

[UTE 90c] FIP Couche Physique en bande de base sur paire torsadée blindée, NF C46-604, Union Technique de l'Electricité, Courbevoie, 1990.

[UTE 91a] FIP Couche Physique en bande de base sur fibre optique, NF C46-607, Union Technique de l'Electricité, Courbevoie, 1991.

[UTE 91b] Bus Eibus: Un réseau pour la gestion technique et administrative des bâtiments, Couche physique, Couche liaison de données, Couche réseau, Couche transport, Couche application, C46-624 à C46-628, Union Technique de l'Electricité, Courbevoie, 1991.

[UTE 92a] FIP Couche Application - Services de Messagerie, NF C46-606, Union Technique de l'Electricité, Courbevoie, 1992.

[UTE 92b] Bus FIP pour échange d'informations entre transmetteurs, actionneurs et automates - Guide de rédaction des normes d'accompagnements, C46-645, Union Technique de l'Electricité, Courbevoie, 1992.

[UTE 92c] Bus Batibus: Un réseau pour la gestion technique et administratrative des bâtiments, Couche physique, Couche liaison de données, Couche application, C46-621 à C46-623, Union Technique de l'Electricité, Courbevoie, 1992.

Documents constructeurs

[ALL 91] Allen Bradley, "Data Highway/Data Highway Plus, communication protocol and command set", Reference manual, document constructeur, 1991.

[APT 92] Aptor, "Factor, manuel de présentation", document constructeur, APTOR, Meylan, 1992.

[BAT 90] Club Batibus, document constructeur n°DBTA 0038A, 1990

[BEN 92] R. Bent, "InterBus-S, System Description", document constructeur, Phoenix Contact, 1992.

[CEG 92] CEGELEC, "FIPCODE Software Version 4, User reference manual", 1992.

[COM 92] Compex, "Les réseaux LAC et LAC2", document constructeur, COMPEX, Pringy, 1992.

[ECH 91] Echelon, "Introducing the LON: the next idea in control technology", document constructeur, 1991.

[EFI 90] Efisystem, "Efiway shortform catalogue", document constructeur, 1990.

[FNE 92] FNE'92, "FAIS Networking Event", document constructeur, 1992.

[FRE 92] L.B. Fredriksson, "A CAN kingdom", KVASER edition, 1992.

[GES 89] Gespac, "Filbus: specifications manual", document constructeur, 1989.

[INT 88] INTEL, "The Bitbus interconnect serial control bus specification", document constructeur, 1988.

[MOD 91] Modicon Modbus Protocol, "Reference guide", MODICON, document constructeur, Industrial Automation Systems, Massachusetts, 1991.

[PHO 92] Phoenix Contact, "The fast approach to a Sensor/Actuator bus device interface", document constructeur, 1992.

[PRO 93] PROFIBUS, "Le standard PROFIBUS", document constructeur, 1993.

[ROS 88] Rosemount, "Les protocoles HART", document constructeur, 1988.

Annexe

Annexe

Caractéristiques techniques et état de l'art de la normalisation des réseaux de terrain

Dans cette annexe, nous allons étudier les réseaux de terrain car c'est dans ces derniers que les aspects temporels sont les plus importants. Nous présenterons la notion de réseau de terrain en nous attachant essentiellement aux architectures de communication utilisées, aux choix techniques et aux aspects temporels. Nous proposerons également un état de l'art concernant la normalisation internationale des réseaux de terrain.

1. Présentation des réseaux de terrain

On appelle couramment réseau de terrain (fieldbus en anglais) un réseau sur lequel on connecte des capteurs, des actionneurs et des automates. Les réseaux de terrain assurent la communication entre les équipements de niveau zéro (les capteurs, les actionneurs) et les équipements de niveau un (les postes opérateurs, les automates, les régulateurs, etc...).

Les réseaux de terrain permettent de remplacer le câblage traditionnel point-à-point entre des équipements d'une installation automatisée. Un premier avantage offert par l'utilisation des réseaux de terrain est donc l'économie de câblage.
Grâce à l'interconnexion des équipements, toute station connectée sur le réseau peut avoir accès à n'importe quelle information détenue par une autre station.

Les réseaux de terrain facilitent la mise en oeuvre et rendent l'installation d'un système automatique beaucoup plus flexible. Comme la plupart des organes de commande sont pratiquement tous numériques et spécialisés (régulateurs, automates, appareils de dialogue-opérateur, etc...), il devient nécessaire de développer des systèmes numériques de communication permettant de répartir la commande de l'installation [GAL 84]. Ainsi, on voit apparaître de nouvelles architectures de systèmes automatisés.

Les réseaux de terrain actuellement disponibles retiennent des caractéristiques différentes concernant le choix du support utilisé, le type de topologie, les méthodes d'accès au médium, les débits de transmission, le coût de connexion, les services et les qualités de service permettant de répondre aux besoins de cohérence temporelle et de cohérence spatiale. Nous allons maintenant présenter quelques uns de ces aspects.

1.1 Réseaux existants

De nombreux réseaux existent tant comme normes que comme produits, citons:

FACTOR [APT 92]
FAIS (Factory Automation Interconnection System) [FNE 92]
LAC [COM 92]
Mini-MAP [MAP 88]
MODBUS [MOD 91]
PROFIBUS (PROcess FIeld BUS) [DIN 90], [KAT 93], [PRO 93]

Concernant les réseaux qui peuvent être considérés comme de véritables réseaux de terrain, on retrouve essentiellement :

BITBUS [INT 88]
Data Highway [ALL 91]
EFIWAY [EFI 90]
FILBUS [GES 89]
FIP (Factory Instrumentation Protocol) [UTE 90], [GAL 84]
HART (Highway Adressable Remote Transducer) [ROS 88]
InterBus-S [PHO 92], [BEN 92]
LONWORKS [ECH 91]

Certains réseaux sont spécialisés pour des application particulières:

- dans l'automobile:
  VAN (Vehicule Area Network)
  CAN (Controller Area Network) [FRE 92]

- en domotique:
  BATIBUS [BAT 90], [UTE 92c]
  Eibus [UTE 91b].

Ainsi parmi ces réseaux, certains sont de véritables réseaux de terrain en terme de services offerts, de coût de raccordement ; d'autres peuvent être utilisés comme réseaux de terrain même s'ils n'ont pas été faits pour cela.

Essayons maintenant de résumer les différentes caractéristiques générales que l'on retrouve dans les réseaux de terrain. Les caractéristiques techniques concernant le nombre maximal de stations que l'on peut connecter, la longueur maximale du médium, les supports de transmission autorisés, le codage des trames utilisé, les références normatives, etc... des différents réseaux énoncés ci-dessus, seront ensuite développées.

1.2 Caractéristiques générales des réseaux

La plupart des réseaux utilisent une architecture de communication ne retenant pas toutes les couches du modèle OSI [ISO 89a]. Certaines couches ne sont pas définies parce qu'elles sont inutiles à ce niveau de la communication et parce qu'elles ralentiraient le transfert des données. Ainsi, si les trames ont une longueur maximale telles qu'elles puissent toujours être transmises en un seul bloc, il n'est alors pas nécessaire d'introduire des mécanismes de segmentation et de réassemblage.

Les réseaux de terrain ne retiennent généralement que les couches physique, liaison de données et application du modèle OSI. Mais tous les réseaux ne suivent pas cette règle, ainsi le réseau CAN utilise uniquement la couche physique et la couche liaison de données du modèle OSI. Par contre le réseau Eibus prend en compte les couches réseau et transport en plus des trois couches du modèle OSI énoncées précédemment ; le réseau LONWORKS, quant à lui, introduit sept couches qui ne sont pas exactement celles du modèle OSI.

Il est possible de classer les réseaux en deux classes : les réseaux basés sur le modèle de communication client/serveur (par exemple les réseaux PROFIBUS et FAIS) et les réseaux basés sur le transfert de variables typées (par exemple les réseaux FIP et BATIBUS). Notons que certains réseaux sont à cheval sur ces deux classes (par exemple le réseau MODBUS), alors que d'autres réseaux penchent plus vers une certaine classe sans qu'il soit possible de les classer complètement dans cette classe (citons le réseau FACTOR qui a tout de même tendance à se rapprocher des réseaux basés sur le transfert de variables typées).

Concernant la méthode d'accès au médium, les réseaux Eibus, BATIBUS, CAN, LONWORKS utilisent la méthode du CSMA/CD ou CSMA/CA. Le réseau de terrain FIP utilise une gestion centralisée, notée arbitre de bus. Les autres réseaux utilisent la technique du modèle maître/esclave où une technique dérivée du modèle maître/esclave (comme par exemple les techniques du multi-maîtres ou encore du maître fixe) [V.AS 93], [ETT 93].

2. Présentation des différents réseaux

Nous allons présenter maintenant les différentes caractéristiques des réseaux cités dans le paragraphe précédent.

2.1 Le réseau BITBUS

BITBUS est un réseau de terrain crée par INTEL et normalisé sous le numéro IEEE 1118. Il peut être utilisé pour la production, le suivi de production, le contrôle/commande, la télésurveillance et la télégestion.

- La topologie employée par la couche physique de BITBUS est celle du bus.

Deux modes de transmission sont permis:

- le mode synchrone qui permet des vitesses de transmission élevées allant de 500 kbit/s à 2,4 Mbit/s, mais dans ce cas, le nombre de noeuds est limité à 28 et la distance à 30 mètres maximum. On utilise dans ce mode de transmission deux lignes, l'une pour transmettre les données et l'autre pour transmettre l'horloge;
- le mode autosynchronisé permet des vitesses de transmission plus faible allant de 62 kbit/s à 375 kbit/s, mais dans ce cas les distances sont plus élevées. Dans ce mode de transmission, on ne véhicule pas l'horloge et le signal est codé en NRZI.

- La couche liaison de données implémente un protocole de type maître/esclave avec une seule station maître. Les services du LLC sont de type 2 et le protocole est de type avec connexion avec contrôle de l'ordre des trames.

- La couche application est basée sur le modèle de communication client/serveur avec un seul client qui est le maître. Toutes les autres stations serveurs sont les esclaves. On retrouve des services de téléchargement, de gestion des tâches et de lecture/écriture.

2.2 Le réseau Data Highway

Data Highway est un réseau de terrain propriétaire développé par Allen-Bradley.

- Concernant la couche physique, Data Highway a retenu la topologie du bus.

• La couche liaison de données est basée sur le principe du maître flottant, c'est-à-dire que chaque noeud a les mêmes droits pour devenir un maître. Ainsi pour transmettre une certaine information, il n'est pas nécessaire de demander à un maître pour avoir le droit d'envoyer une information.

2.3 Le réseau FACTOR

FACTOR est un réseau local industriel défini par la société APTOR, permettant de relier entre eux différents équipements d'automatismes ainsi que des calculateurs.

• La couche physique du réseau FACTOR permet de mettre en oeuvre une structure en bus, en étoile ou encore une combinaison des deux.
Au niveau de la couche physique, on peut retrouver du câble coaxial, de la paire torsadée ainsi que de la fibre optique. La longueur du médium doit être inférieur à 2800 mètres si c'est le CSMA/CD est utilisé et inférieur à 10000 mètres si c'est le CSMA/DCR qui est utilisé. On retrouve différentes vitesses autorisées: 2 Mbit/s, 10 Mbit/s ou encore 100 Mbit/s, ce qui permet d'être conforme au standard FDDI (Fiber Distributed Data Interface).

• Concernant la couche liaison de données, FACTOR est conforme au standard 802.3 qui utilise, pour l'accès au support de transmission, la technique du CSMA/CD (Carrier Sense Method Access with Collision Detection). Dans le cas où l'on retrouve de fortes contraintes de temps, FACTOR utilise la technique du CSMA/DCR (Carrier Sense Method Access with Deterministic Collision Resolution) qui supprime l'aspect aléatoire du CSMA/CD.
Les services du LLC sont de type 2 et le protocole est de type avec connexion.

• Les services et protocoles des couches réseau et transport sont du type avec connexion.

2.4 Le réseau FAIS

FAIS (Factory Automation Interconnection System) est un réseau d'origine japonaise.

• La couche physique est conforme au standard IEEE 802.4, la topologie retenue est celle du bus.

• La couche liaison de données de FAIS est basée sur la notion de jeton circulant entre les différents équipements connectés sur le réseau.

• Au niveau de la couche application, FAIS reprend les services et les protocoles définis dans la norme MMS.

2.5 Le réseau FIP

Les principales caractéristiques du réseau de terrain FIP ont déjà été étudiées en détail dans le chapitre 4.

• Les caractéristiques de la couche physique du réseaux de terrain FIP sont les suivantes [UTE 90c], [UTE 91a]:

- Nombre maximum de stations: 256,
- Médium: paire torsadée ou fibre optique,
- Longueur maximum: 2000 mètres,
- Codage: Manchester II,
- Topologie: bus pour la paire torsadée et en étoile multiples actives ou passives, pour la fibre optique,
- Débit:
    + 31,25 kbit/s pour les applications couvrant de longues distances et possédant de nombreuses connexions,
    + 1 Mbit/s et 2,5 Mbit/s pour les applications géographiquement concentrées et demandant de bonnes performances,
    + 5 Mbit/s uniquement sur fibre optique.

Les différentes normes existantes de FIP sont présentés dans le schéma suivant:

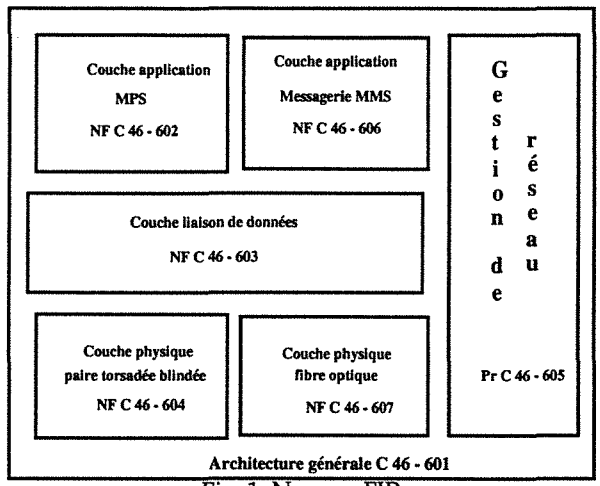


Fig. 1: Normes FIP

Un fascicule documentaire présentant un cadre rédactionnel pour la spécification des normes d'accompagnement du bus FIP est référencé dans [UTE 92b].

2.6 Le réseau InterBus-S

InterBus-S est un réseau de terrain développé par Phoenix Contact.

- La couche physique d'InterBus-S a retenu une topologie en anneau.

- La méthode d'accès au médium repose sur le principe maître/esclave.

- Une partie des services de communication de la couche application d'InterBus-S, appelée PMS (Peripherals Message Specification) est compatible avec les services de communication FMS de PROFIBUS.
Des mécanismes de synchronisation peuvent être mis en oeuvre grâce à l'ajout d'un signal d'horloge supplémentaire.

2.7 Le réseau LAC

LAC est un réseau développé par COMPEX. On retrouve le réseau LAC dans les chaînes de fabrication automatisée, dans les ateliers, dans les applications minière ainsi que dans les systèmes de contrôles et de tests.

- La topologie est un bus et la méthode d'accès utilisée est le CSMA CA/CD.

Au niveau du médium, on retrouve de la paire torsadé blindée, la longueur maximale est de 8 km pour LAC et de 3 km pour LAC2. Le débit maximum permis pour LAC est de 50 kBauds et de 250 kBauds pour LAC2.

• La couche liaison de données du réseau LAC est construite sur le modèle maître/esclave.

• Dans la couche application, on retrouve des fonctions d'émission/réception, d'acquittements, de téléchargement, de télédiagnostic, de diffusion, de mise à l'heure et de lecture mémoire.

2.8 Le réseau LONWORKS

Les LON (Local Operating Network) qui sont conformes au profil LONTALK, développé par ECHELON, sont appelés des LONWORKS. Les principales applications de LONWORKS ont été développées dans le domaine de la domotique.
Sur le LON, on retrouve un ensemble de noeuds qui sont composés en général uniquement de capteurs et d'actionneurs qui communiquent entre eux sans système central.

• La topologie retenue par LONWORKS est le bus. Plusieurs médiums de communication peuvent être utilisés: le câble coaxial, la paire torsadé, la fréquence radio, l'infrarouge, la fibre optique et le courant porteur.

• La couche liaison de données du réseau LONWORKS utilise comme protocole d'accès au médium le CSMA/CD.

2.9 Le réseau PROFIBUS

• Les caractéristiques de la couche physique du réseaux PROFIBUS (PROcess FIeld BUS) sont les suivantes [DIN 90]:

- Nombre de stations maîtres: 32,
- Nombre de stations esclaves: 127,
- Médium: paire torsadée,
- Longueur: 1200 mètres,
- Codage: NRZ,
- Débit: 9,6 ; 19,2 ; 93,75 ; 187,5 et 500 kbit/s.

La topologie retenue est celle du bus.

• Dans PROFIBUS, les stations maîtres sont appelées les stations actives et sont typiquement composés d'automates, de contrôleurs numériques et de contrôleurs de cellules. Entre les stations maîtres, la communication se fait par l'intermédiaire d'un anneau logique avec passage d'un jeton circulant.
Les stations esclaves sont appelées les stations passives ; elles sont principalement composées de capteurs, d'actionneurs et de transmetteurs.

Le Medium Access Control de PROFIBUS repose sur une méthode hybride. Les stations maîtres utilisent la méthode du jeton circulant pour communiquer entre elles. Chaque station maître possède alors le jeton pendant une durée donnée. Durant cette période elle peut dialoguer avec chacune des stations esclaves (communication de type maître/esclave) ainsi qu'avec les autres stations maîtres (communication de type maître/maître). Lorsque la période est échue, la station maître doit passer le jeton à la station maître suivante. Le passage du jeton d'une station maître vers une autre station maître se fait selon une séquence figée définie lors de la configuration. Dans le réseau PROFIBUS, la période maximale de circulation du jeton entre les différentes stations maîtres est paramétrable.

Un exemple de communication entre trois stations maîtres et neuf stations esclaves peut se schématiser de la manière suivante:

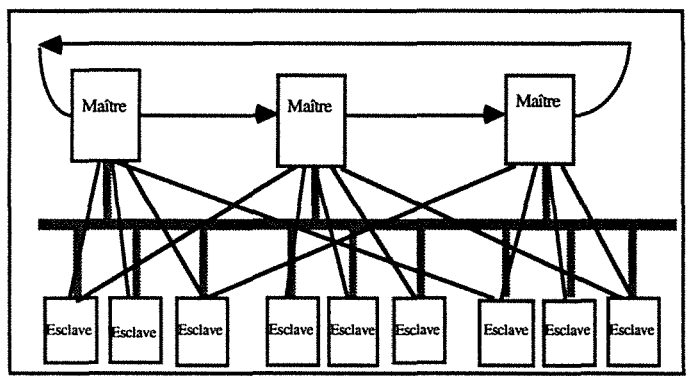


Fig. 2: Exemple de communication de stations maîtres et de stations esclaves dans PROFIBUS

Dans le réseau PROFIBUS, si deux stations maîtres différentes souhaitent consommer une même information en provenance d'une station esclave, elles n'auront pas nécessairement la même information puisque chacune des deux stations maîtres consulte la station esclave uniquement lorsqu'elle possède le jeton, donc à des instants différents. Si la station esclave est, par exemple, un capteur alors entre deux instants différents, l'information en provenance du capteur aura eu le temps d'être modifiée. On ne sera donc pas capable d'élaborer des services de cohérence spatiale avec le réseau PROFIBUS. Les seuls services temporels que l'on retrouve concernent le temps alloué à une

station maître pour mettre en oeuvre ses échanges avec d'autres stations maîtres ou esclaves.
La norme PROFIBUS désigne la couche liaison de données sous le terme de FDL (Fieldbus Data Link). Les services du LLC sont de type 3 et le protocole est de type sans connexion, c'est-à-dire qu'une station active émet un message à un groupe de stations ou à toutes les stations et ceci sans accusé de réception.

- La couche application de PROFIBUS, inspirée de MMS [ISO 89b], est composée de FMS (Fieldbus Message Specification) qui décrit les objets de communication et les services de niveau application ainsi que du LLI (Lower Layer Interface) qui permet d'adapter les services de niveau application à la couche liaison de données.
Dans FMS, on distingue les services avec et sans accusé de réception. La transmission des services sans accusé de réception peut se faire selon différents degrés de priorité.
La partie LLI assure le contrôle des flux et des connexions ainsi que le transfert des services FMS vers la couche liaison de données.

- Dans PROFIBUS, on retrouve des services de gestion de réseau permettant :
  - la gestion de configuration pour charger et modifier les accès aux variables ainsi que l'identification des hiérarchies de communication,
  - la gestion de contexte permettant l'établissement et l'interruption d'une connexion,
  - la gestion de défauts du réseau.

2.10 Le réseau CAN (Controller Area Network)

Le réseau CAN est un réseau développé pour répondre aux besoins du secteur de l'automobile. CAN implémente uniquement les couches physique et liaison de données du modèle OSI.

- Le réseau CAN utilise une topologie en bus. Actuellement, les micro-contrôleurs CAN permettent de mettre en oeuvre des mécanismes de synchronisation.

- La méthode d'accès mise en oeuvre par la couche liaison de données du réseau CAN est le CSMA/CD.

2.11 Le réseau VAN (Vehicule Area Network)

Le réseau VAN permet d'interconnecter entre eux différents équipements d'une automobile (autoradio, climatisation, ordinateur de bord, rétroviseurs, suspension, ABS, etc...).

- La couche physique de VAN a retenue la topologie du bus.

- La couche liaison de données de VAN est basée sur une architecture de communication de type multi-maîtres/multi-esclaves.

- La couche application est en cours d'élaboration.

2.12 Tableaux récapitulatifs des différentes caractéristiques techniques

Nous allons récapituler les différentes caractéristiques de l'ensemble des réseaux introduits, dans un tableau comportant les rubriques suivantes:

- référence normative du réseau,
- topologie utilisée,
- support de transmission,
- codage,
- longueur maximale supportée par le réseau sans mettre en oeuvre de répéteurs,
- débit du réseau,
- mécanisme utilisé pour accéder au médium de communication,
- nombre maximal de stations qu'il est possible de connecter sur le réseau,
- énumération des couches du modèle OSI utilisées.

	FIP	**PROFIBUS**	**LONWORKS**
Norme	NF C46-601 à NF C46-607	DIN 19245-1 DIN 19245-2	ECHELON
Topologie	Bus	Bus	Bus
Support de transmission	Paire torsadée, fibre optique	Paire torsadée	Paire torsadée, fibre optique, câble coaxial, fréquence radio, infrarouge
Codage	Manchester 2	NRZ	
Longueur maximale (sans répéteur)	2000 mètres	1200 mètres	
Débit	31,25 kbit/s, 1 Mbit/s, 2,5 Mbit/s et 5 Mbit/s	9,6 ; 19,2 ; 93,75 ; 187,5 et 500 kbit/s	75 kbit/s à 1,25 Mbit/s
Accès au médium	Arbitre de bus	Passage de jeton entre stations maîtres, interrogation des stations esclaves par les stations maîtres	CSMA/CD
Nombre maximal de stations	256	32 maîtres 127 esclaves	32000
Couches OSI	1, 2 et 7	1, 2 et 7	les 7 couches

	BITBUS	**FACTOR**	**FAIS**
Norme	INTEL	APTOR	Japonaise
Topologie	Bus	Bus ou étoile	Bus
Support de transmission	Paire torsadée	Câble coaxial, paire torsadée, fibre optique	Fibre optique, paire torsadée
Codage	NRZI		
Longueur maximale (sans répéteur)	13,2 km	2800m (CSMA/CD) 10km(CSMA/DCR)	500 m
Débit	62kbit/s à 2,4Mbit/s	2, 10 et 100 Mbit/s	5 et 10 Mbit/s
Accès au médium	Maître/esclave	CSMA/CD et CSMA/DCR	Jeton ISO 8802-4
Nombre maximal de stations	250	128	
Couches OSI	1, 2 et 7	1 à 4 et 7	1, 2 et 7

	LAC	**HART**	**VAN**
Norme	COMPEX	ROSEMOUNT	ISO 11519-2
Topologie	Bus	Bus	Bus
Support de transmission	Câble coaxial, paire torsadée	Paire torsadée	Paire torsadée, fibre optique
Codage		Bell 202	Manchester
Longueur maximale (sans répéteur)	8 km	3000 m	
Débit	50 et 250 kbauds	1200 bauds	1 Mbit/s
Accès au médium	CSMA CA/CD	Maître/esclave	Multi-maîtres multi-esclaves
Nombre maximal de stations	252	15	
Couches OSI		1, 2 et 7	1, 2 et 7

	Data Highway	**CAN**	**InterBus-S**
Norme	ALLEN-BRADLEY	ISO 11519-1	PHOENIX CONTACT
Topologie	Bus	Bus	Anneau
Support de transmission	Paire torsadée		Paire torsadée, fibre optique, infrarouge
Codage	RS-232-C ou RS-422-A	NRZ	RS-485
Longueur maximale (sans répéteurs)	3000 m	40 m	400 m
Débit	57,6 kbauds	125kbit/s à 1Mbit/s	500 kbit/s
Accès au médium	Maître flottant	CSMA/CD	Maître/esclave
Nombre maximal de stations	255		256
Couches OSI	1, 2 et 7	1 et 2	1, 2 et 7

	EFIWAY	**FILBUS**
Norme		
Topologie	Anneau	Bus
Support de transmission	Fibre optique	Paire torsadée
Codage		NRZI
Longueur maximale (sans répéteur)	52 km	
Débit		375 kbit/s
Accès au médium	Accès déterministe par tranches temporelles	Maître/esclave
Nombre maximal de stations	64	254
Couches OSI	1, 2 et 7	1, 2 et 7

3. Normalisation des réseaux industriels

Parler de normalisation en communication revient souvent à considérer une définition de services et une définition de protocoles auquel on associe les notions de normes d'accompagnement, de profils, de tests de conformité, puis de tests d'interopérabilité. Ces problèmes sont actuellement traités séquentiellement dans l'ordre dans lequel ils apparaissent.
Beaucoup de comités définissent de nouvelles architectures et de nouveaux protocoles en partant de besoins sectoriels. On se dirige donc tout droit vers autant de réseaux de terrain et de réseaux temps critique qu'il existe de domaines d'application.

La normalisation internationale est un aspect très important dans les réseau de terrain. Ainsi, de nombreux organismes nationaux, européens et mondiaux se préoccupent de la normalisation des systèmes de communication, ce qui crée une situation internationale complexe, dans laquelle les actions de tous ces organismes internationaux ne sont pas toujours très coordonnées, d'autant plus que cette multiplicité sert les intérêts des acteurs qui ne souhaitent pas toujours cette coordination.

3.1 Acteurs intervenant dans le processus de normalisation

Les principaux comités de travail oeuvrant dans la mise au point de la norme réseau de terrain sont:

3.1.1 Au niveau international

- ISO TC 184 SC5 WG2 : normalisation d'interfaces, de protocoles, de formats de données, de structure de messages pour l'interconnexion des équipements et des systèmes dans des usines automatisées. On y trouve deux sous-groupes :

  - ISO TC 184 SC5 WG2 TCCA (groupe conjoint ISO/CEI) : étude des exigences et des recommandations pour les architectures de réseaux permettant à la fois la transmission de donnés avec et sans contraintes de temps strictes, dans des domaines d'application tels que la production manufacturière,
  - ISO TC 184 SC5 WG2 MMS : définition de services permettant l'échange de messages entre machines programmables (automates, robots, machines outils ...).

- ISO TC 184 SC1 WG3 : définition de normes d'accompagnement à MMS destinées à compléter la norme de base et à définir des services spécifiques aux besoins des machines de commande numérique.

- ISO TC 184 SC2 WG6 : définition de normes d'accompagnement à MMS destinées à fournir des services spécifiques aux besoins des robots.

- ISO TC 72 WG13 : définition des besoins propres de l'industrie textile et choix du réseau de terrain y répondant le mieux, sachant qu'elle ne peut attendre la parution de la norme internationale en cours d'élaboration à la CEI SC 65C.

- ISO TC 184 SC1 : communications pour la commande numérique de machines et de robots.

- CEI SC 65C et CEI SC 65A : normes sur les bus de terrain, les blocs fonctionnels et les architectures de communication. On y trouve plusieurs sous-groupes:

  - CEI 65C WG6: ce groupe est chargé de définir une norme unique répondant à tous les besoins des réseaux de terrain,
  - CEI SC 65C WG1 : définition de normes d'accompagnement à MMS destinées à fournir des services spécifiques aux besoins du process control ; ce travail a été confié à l'ISA SP 72,
  - CEI SC 65A WG 6 TF 7 et CEI SC 65B : définition de normes d'accompagnement à MMS destinées à fournir des services spécifiques aux besoins des automates programmables.

- CEI SC 65 WG6: ce groupe, nouvellement créé, a reçu pour mission de définir un modèle de blocs fonctionnels permettant de décrire le comportement des équipements raccordés vis-à-vis du réseau.

- ISO/CEI JTC 1: tout ce qui concerne la définition des techniques de l'information :

  - ISO/CEI JTC1 SC21 : définir l'architecture de communication et les couches hautes pour l'interconnexion des systèmes de communication,
  - ISO/CEI JTC 1 SC 6 : prise en charge des couches basses.

- on retrouvera à l'ISO des groupes se préoccupant de communication pour les besoins suivants :

  + applications des ordinateurs à la construction navale (ISO TC8 SC15),
  + équipements électroniques dans l'agriculture (ISO TC 23),
  + équipement pour l'industrie pétrolière (ISO TC67),
  + équipement pour l'industrie minière (ISO TC 82).

3.1.2 Au niveau européen

- CENELEC BT TF 62-6 : recherche des besoins européens pour les bus de terrain, examen des travaux de la CEI y répondant et proposition de solutions pour les besoins non satisfaits ; on soulignera que les besoins des Européens diffèrent des besoins des Américains.

- CEN TC 247 : bus pour la gestion technique des bâtiments.

- CEN TC 251 GT3 : bus dans le domaine paramédical et hospitalier.

- CLC TC 105 : systèmes électroniques domestiques.

- EWOS : étude des profils sur divers sujets de l'OSI.

Par ailleurs, d'importantes contributions européennes sont préparées par divers projets ESPRIT et EUREKA, dont FICIM, FIELBUS, DIAS, INCA... Ainsi,

- EUREKA EU 68 est un projet qui s'est donné pour objectif de définir des produits compatibles réseau de terrain ainsi que les outils nécessaires à cet objectif,
- ESPRIT-FICIM est un projet d'origine germanique qui vise à la définition de l'intégration des réseaux de terrain dans une architecture CIM. Ce projet est maintenant terminé.

3.1.3 Au niveau national

- UTE CEF 44 : miroir français de la CEI TC 44.

- UTE CEF 65C : miroir de la CEI SC 65C.

- UTE CEF 65C GE6 : établissement des normes FIP.

- AFNOR/UTE SMI : préparation au niveau français des travaux pour l'ISO/CEN et la CEI/CENELEC sur la messagerie industrielle.

- UNM 89 B : machines textiles.

- Groupe Domotique.

Notons que le DKE et l'UTE sont deux organismes nationaux (respectivement allemand et français) qui ont été les vecteurs normatifs initiaux de PROFIBUS et de FIP. Ils constituent en fait les références techniques vis-à-vis de la CEI.

3.1.4 Autres organisations

Si la normalisation est l'oeuvre de l'ISO et de la CEI, il ne faut pas oublier les travaux effectués par les:

- organismes ou sociétés savantes tels que ISA, IEEE qui se voient souvent confier des tâches d'écriture de normes. L'ISA SP50 est un comité de l'"Instrument Society of America" qui est mandaté par la normalisation américaine pour définir le "Fieldbus",

- organismes d'associations d'industriels tels que NEMA, EIA qui ont aussi leurs groupes d'experts et parfois leur "task force" qui proposent également des normes,

- projets "prénormatifs" qui regroupent des industriels d'un ou de plusieurs pays pour démontrer l'intérêt d'un nouveau système et ceci en général dans un but normatif. On peut citer ici le groupe MAP dans le passé, le groupe FAIS et le groupe IFC,

- IFC (International Fieldbus Consortium) qui regroupe 60 membres et qui vise à démontrer la validité des standards CEI dans le cadre de tests industriels.

3.2 Normalisation des réseaux de terrain à la CEI

Concernant la normalisation des couches physique, liaison de données et application ainsi que les normes d'accompagnement par la CEI, l'état de la situation est la suivante:

- couche physique: un consensus s'est formé autour d'un document unique autorisant trois vitesses: 31,25 kbit/s, 1 Mbit/s et 2,5 Mbit/s. La norme est actuellement publiée en tant que IS. Les travaux ont débuté et sont bien avancés pour la définition d'un média fibre optique et d'une version radio,

- couche liaison de données: après des travaux très exhaustifs sur les solutions potentielles, les groupes ISA/CEI ont approuvé le choix d'un document de synthèse baptisé pour l'ISA 359/360 et définissant des services et protocoles. Un document suffisamment stable pour émettre un DIS pour vote national et permettre des implémentations prototypes est en voie d'achèvement,

- couche application: devant les difficultés et l'absence de progression des travaux, les comités ISA et CEI ont décidé de former fin 1990 un comité

d'édition commun. Le comité d'édition s'est doté d'une méthode de travail et d'un planning conduisant à la réalisation d'une spécification. Ces travaux sont dans une passe difficile compte tenu de l'ampleur du problème, de la tendance à la réécriture des standards existants et de l'absence de volonté de coopération de certains membres. Les travaux sont en retard et la qualité des documents ne répond pas toujours aux attentes des utilisateurs,

- normes d'accompagnement: des travaux menés par le SP50 ont défini des blocs fonctionnels standards permettant la description des équipements banalisés dans le cadre d'applications simples de contrôle des procédés.

Table des matières

Introduction générale 9

Chapitre 1: La communication dans les Systèmes Automatisés 13

1. Introduction 15
2. Modèles d'architectures fonctionnelles 17
   - 2.1 Le modèle NBS 18
   - 2.2 Architecture pour les processus continus 18
   - 2.3. Modèle trois axes 19
3. Les réseaux dans les Systèmes Automatisés 20
4. Architecture fonctionnelle, architecture matérielle et architecture opérationnelle 22
5. Modèles de communication dans les architectures opérationnelles 24
   - 5.1 Modèle Producteur/Consommateur 25
   - 5.2 Modèle Producteur/Consommateurs 25
   - 5.3 Modèle Producteurs/Consommateurs 27
   - 5.4 Le modèle Client/Serveur 29
   - 5.5 Le modèle Client/Multiserveurs 31
   - 5.6 Le modèle Multiclients/Multiserveurs 33
6. Conclusion 35

Chapitre 2: Spécifiaction des contraintes temporelles 37

1. Introduction 39
2. Notions de contraintes temporelles 40
3. Spécification des contraintes temporelles 42
    3.1 L'algèbre d'intervalles 43
    3.2 Remarques 44
4. Fenêtres temporelles dans la communication 44
    4.1 Notations pour les fenêtres temporelles 49
    4.2 Fenêtres temporelles d'activité 50
        4.2.1 Notations 50
        4.2.2 Enchaînement des fenêtres temporelles d'activité 51
    4.3 Fenêtres temporelles de la validité des variables 56
        4.3.1 Notations 56
        4.3.2 Règles d'enchaînement 57
    4.4 Configuration des fenêtres temporelles 59
5. Conclusion 63

Chapitre 3: Modélisation et validation des mécanismes de fenêtres temporelles 65

1. Introduction 67
2. Modélisation des fenêtres temporelles 67
    2.1 Présentation des différentes sous-machines d'états 70
        2.1.1 Les réseaux de Nutt 70
        2.1.2 Passage vers l'état Attente et de l'état Attente sur lui-même 72
        2.1.3 Passage de l'état Attente vers l'état E-Faux 72
        2.1.4 Passage de l'état Attente vers l'état E-Vrai et de l'état E-Vrai vers lui-même 73
        2.1.5 Passage de l'état E-Vrai vers l'état Expiration1 74
        2.1.6 Passage de l'état Expiration1 vers l'état Expiration2 et vers l'état E-Faux 75
    2.2 Machine d'états générale 76
3. Validation et implantation de la machine d'états générale 78
    3.1 Etape de validation 78
    3.2 Implantation de la machine d'états générale 82
4. Application des mécanismes temporels étudiés aux modèles de communication 87
    4.1 Modèle Client(s)/Serveur(s) "temporel" 87
        4.1.1 Mécanismes temporels de l'entité client 88
        4.1.2 Mécanismes temporels de l'entité serveur 89
    4.3 Liens entre les modèles de communication Client(s)/Serveur(s) et Producteur(s)/Consommateur(s) 90
5. Conclusion 90

Chapitre 4: Application au réseau de terrain FIP 93

1. Introduction 95
2. Le réseau de terrain FIP 95
3. Statuts temporels du réseau de terrain FIP 100
    3.1 Présentation des statuts de rafraîchissement et de promptitude ainsi que de leurs caractéristiques 100
        3.1.1 Statut de rafraîchissement 101
        3.1.2 Statut de promptitude 101
        3.1.3 Caractéristiques des statuts de rafraîchissement et de promptitude 101
    3.2 Introduction d'un nouveau statut temporel: le statut de fraîcheur-de-consommation 103
    3.3 Description des protocoles 103
4. Application des mécanismes de fenêtres temporelles aux statuts temporels 105
    4.1 Statuts temporels du réseau de terrain FIP 105
        4.1.1 Statut de rafraîchissement 106
        4.1.2 Statut de promptitude 107
        4.1.3 Remarques 108
    4.2 Cas du statut de fraîcheur-de-consommation 109
    4.3 Analyse des statuts temporels dans FULLFIP 110
5. Liens entre les mécanismes temporels de FIP et la machine d'états générale 114
    5.1 Caractère asynchrone 114
    5.2 Caractère synchrone 115
    5.3 Caractère ponctuel 116
6. Besoins temporels concernant les listes de variables 117
    6.1 Notion de buffer privé/public 118
    6.2 Représentation des machines d'états temporelles pour les listes de variables 119
    6.3 Statut de cohérence spatiale 121
7. Conclusion 122

Conclusion générale 125

Bibliographie 129

Annexe : Caractéristiques techniques et état de l'art de la normalisation des réseaux de terrain 145

1. Présentation des réseaux 147
    1.1 Réseaux existants 148
    1.2 Caractéristiques générales des réseaux 149
2. Présentation des différents réseaux 150
    2.1 Le réseau BITBUS 150
    2.2 Le réseau Data Highway 150
    2.3 Le réseau FACTOR 151
    2.4 Le réseau FAIS 151
    2.5 Le réseau FIP 152
    2.6 Le réseau InterBus-S 153
    2.7 Le réseau LAC 153
    2.8 Le réseau LONWORKS 153
    2.9 Le réseau PROFIBUS 154
    2.10 Le réseau CAN (Controller Area Network) 156
    2.11 Le réseau VAN (Vehicule Area Network) 157
    2.12 Tableaux récapitulatifs des différentes caractéristiques techniques 157
3. Normalisation des réseaux industriels 161
    3.1 Acteurs intervenant dans le processus de normalisation 161
        3.1.1 Au niveau international 161
        3.1.2 Au niveau européen 163
        3.1.3 Au niveau national 163
        3.1.4 Autres organisations 164
    3.2 Normalisation des réseaux de terrain à la CEI 164

Table des matières 167

INSTITUT NATIONAL
POLYTECHNIQUE
DE LORRAINE

AUTORISATION DE SOUTENANCE DE THESE DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

o0o

VU LES RAPPORTS ETABLIS PAR :
Monsieur ELLOY Jean-Pierre, Maitre de Conférence, Ecole Centrale Nantes,
Monsieur JUANOLE Guy, Professeur, L.A.A.S. Toulouse.

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur LORENZ Pascal

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :
"Le temps dans les architectures de communication : application au réseau de terrain FIP".

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, **28 Juin 1994**
Le Président de l'I.N.P.L.,
M. LUCIUS

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE NANCY

NANCY BRABOIS
2, AVENUE DE LA
FORET-DE-HAYE
BOITE POSTALE 3
F - 5 4 5 0 1
VANDŒUVRE CEDEX

TEL. 33/83.59.59.59
FAX. 33/83.59.59.55

Résumé

Les RLI (Réseaux Locaux Industriels) permettent de faire communiquer les divers équipements d'un Système Automatisé. Il existe plusieurs niveaux de RLI parmi lesquels on trouve les réseaux de terrain. Ces derniers offrent la possibilité de mettre en oeuvre des échanges d'informations nécessitant de respecter de fortes contraintes de temps.

Différentes techniques de spécification peuvent être utilisées pour exprimer les contraintes de temps dans les systèmes de communication. Nous avons choisi de retenir l'algèbre d'intervalles qui permet de représenter sous forme de fenêtres temporelles les différentes contraintes de temps intervenant dans la communication et d'exprimer les différentes relations existantes entre ces fenêtres temporelles.

Pour exprimer les contraintes de temps intervenant dans la communication entre deux entités, nous avons introduit des fenêtres temporelles de production, d'émission, de réception et de consommation, ainsi que pour représenter la durée de vie des variables. Un ensemble de règles permettant de relier les différentes fenêtres temporelles entre elles ont été développées. A partir de ces règles, nous avons déduit une machine d'états générale qui génère des statuts temporels permettant de savoir si les contraintes de temps imposées à une étape de la communication ont été respectés ou non. Cette dernière a été validée et implantée au dessus de TCP/IP.
Nous avons ensuite montré comment les mécanismes temporels introduits permettent de retrouver et de compléter les différents mécanismes temporels présents dans la couche application du réseau de terrain FIP.

Mots clés: Réseau de terrain, Temps réel, Architecture de communication, Fenêtres temporelles, Mécanismes temporels, Contraintes de temps, Modèles de communication, Machines d'états.

Abstract

The LAN (Local Area Networks) allow the communication between the various devices of an Automated System. There are several levels of LAN among which we find fieldbuses. These latter offer the possibility to exchange information which require to meet strong temporal constraints.

Several specification techniques can be used to express the temporal constraints in the communication systems. We have chosen to use the interval algebra which allows to represent with time windows the different temporal constraints which occur during the communication and to express the different relations existing between these temporal windows.

To express the temporal constraints in the communication between two entities, we have chosen to introduce production, sending, receiving and consuming time windows, as well as to represent the life time. A lot of rules enable to link the different developed time windows. From theses rules, we have deduced a general state machine which generate temporal statuses allowing to know if the temporal constraints imposed to a step of the communication have been respected or not. This latter has been validated and is implemented above TCP/IP.
Then we have shown how the introduced temporal mechanisms allow to find again and to complete the different temporal mechanisms in the FIP fieldbus application layer.

Key words: Fieldbus, Real-time, Communication architecture, Time window, Temporal mechanisms, Temporal constraints, Communication models, States machines.