



HAL
open science

Contribution à la modélisation des systèmes multimédias et hypermédias

Patrick Sénac

► **To cite this version:**

Patrick Sénac. Contribution à la modélisation des systèmes multimédias et hypermédias. Réseaux et télécommunications [cs.NI]. INSA de Toulouse, 1996. Français. NNT: . tel-00139857

HAL Id: tel-00139857

<https://theses.hal.science/tel-00139857>

Submitted on 3 Apr 2007

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

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

Année 1996

THÈSE

présentée au

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTÈMES DU CNRS

en vue de l'obtention du titre de

DOCTEUR DE L'UNIVERSITÉ PAUL SABATIER

Spécialité : Informatique

par

Patrick SÉNAC

CONTRIBUTION A LA MODÉLISATION DES SYSTÈMES MULTIMÉDIAS ET HYPERMÉDIAS

Soutenue le 12 Juin 1996

devant le jury composé de :

MM.	Claude	BETOURNÉ	Président
	Richard	CASTANET	Rapporteur
	Alain	COSTES	Examineur
	Jean-Pierre	COURTIAT	Examineur
	Michel	DIAZ	Directeur de Thèse
	Jean-Louis	FRÉSON	Examineur
	Claude	GIRAULT	Rapporteur
	Eric	HORLAIT	Rapporteur
	Alain	LÉGER	Examineur

Thèse préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 7
Avenue du Colonel Roche, 31077 Toulouse Cedex, France

A mes Parents, à Christine et Florence...

Avant Propos

Si vous me permettez la métaphore je comparerais le travail développé dans ce mémoire, à un composant d'un vaste hypertexte dont les ramifications prolongent et approfondissent le sens de cette thèse. Aussi de même que la signification d'un hypertexte ne se limite pas à celle d'un de ses composants mais fait intervenir tout un réseau de signifiants, ce travail est indissociable de tous les soutiens dont j'ai bénéficié afin de le mener à son terme.

Je voudrais en tout premier lieu souligner le rôle déterminant qu'ont joué Messieurs Alain COSTES, Directeur du LAAS et Pierre SINTÉS, Directeur de l'ENSICA de 1986 à 1994 et actuel Directeur de l'ENSTA, dans la réalisation de cette thèse. Sans leur esprit d'ouverture et leur appui ce travail n'aurait pu voir le jour. Je leur témoigne une reconnaissance et un attachement indéfectibles.

J'exprime ma profonde gratitude à Monsieur Jean-Louis FRÉSON, Directeur de l'ENSICA et à Monsieur Patrick HÉBRARD Directeur des Études et de la Recherche de l'ENSICA pour la chaleureuse et réconfortante confiance qu'ils m'ont constamment adressée. Je tiens particulièrement à rendre hommage à la très efficace et très attachante équipe de Direction qu'ils constituent.

Monsieur Michel DIAZ, Directeur de Recherche au CNRS et responsable du Groupe Outils et Logiciels pour la Communication du LAAS du CNRS, m'a initié à la recherche puis guidé dans mes travaux avec une disponibilité et une ferveur de chaque instant. Il m'a fait découvrir le bonheur, la joie et l'exaltation que l'on puise dans une activité de recherche. Je le remercie très chaleureusement pour m'avoir accueilli amicalement dans son équipe, et pour avoir été pour moi une inépuisable source d'inspiration.

Les idées développées dans cette thèse ont vu le jour au cours d'un séjour que j'ai réalisé dans le groupe de recherche du Professeur Pravin Varaiya de l'Université de Californie à Berkeley. Je remercie très cordialement le Professeur Pravin Varaiya pour m'avoir accueilli dans son équipe et pour m'avoir offert un environnement de travail ayant permis l'éclosion de ce travail de recherche.

J'exprime mes plus vifs remerciements à Messieurs:

Claude BETOURNÉ, Professeur à l'Université Paul Sabatier,
Richard CASTANET, Professeur à l'Université de Bordeaux I,
Alain COSTES, Directeur du LAAS, Président de l'Institut National Polytechnique de Toulouse,
Jean-Pierre COURTIAT, Chargé de Recherche au CNRS,
Michel DIAZ - Directeur Recherche au CNRS,
Jean-Louis FRÉSON, Directeur de l'École Nationale d'Ingénieurs de Constructions Aéronautiques
Claude GIRAULT, Professeur à l'Université Paris VI
Eric HORLAI, Professeur à l'Université Paris VI
Alain LÉGER, Chef de Département au CCETT,

pour l'honneur qu'ils me font de participer à la Commission d'Examen. Je remercie particulièrement Messieurs Richard CASTANET, Claude GIRAULT et Eric HORLAI pour avoir accepté la charge de rapporteur. J'adresse mes plus chaleureux remerciements à Monsieur Claude BETOURNÉ qui m'a brillamment révélé l'informatique lorsque j'étais jeune étudiant et qui est à l'origine de ma conversion à cette science. En me faisant l'honneur de présider cette Commission d'Examen il ajoute au statut de président une portée symbolique à laquelle je suis très sensible.

Cette thèse a été réalisée dans le cadre du projet national CESAME (Conception formELLE de Systèmes distribués coopératifs haut-débit MultimEdias) associant le CNET et le CNRS. Les travaux décrits dans ce mémoire ont été grandement dynamisés par la saine émulation et l'échange stimulant d'idées qui se sont exprimés dans ce projet de recherche. Je soulignerais particulièrement les discussions et débats d'idées extrêmement enrichissants que j'ai eus avec Messieurs Richard CASTANET, Jean-Pierre COURTIAT, Eric HORLAI et Alain LÉGER.

Fabrice, Laurent, Pierre, René et Yves, mes collègues et amis du Département de Formation et de Recherche en Mathématique et Informatique de l'ENSICA en développant autour de moi une amitié et une solidarité sans faille et exemplaires ont grandement facilité mon travail de recherche. Merci mes très chers amis.

Je voudrais également remercier tous mes collègues de l'ENSICA et du LAAS qui m'ont assuré de leur soutien cordial, ainsi que l'ensemble du personnel technique et administratif de ces deux établissements pour leur support dans la réalisation de mon travail de thèse.

Mes Parents, mon épouse Christine et ma fille Florence ont certainement été ma principale source de motivation pour réaliser ce travail, je leur dédie cette thèse. Si mon nom apparaît sur ce mémoire c'est aussi le leur.....

Table des Matières

Chapitre 1 : Introduction	1
Chapitre 2 : Problématique de la Synchronisation Multimédia	9
1. Introduction.....	9
2. Organisation du Chapitre.....	10
3. Le Paradigme de la Régie de Production.....	10
3.1. Les Limitations du Paradigme de la Régie de Production.....	12
4. Les Unités d'Information.....	13
4.1. De l'intérêt du Concept d'Unité d'Information.....	14
5. La Notion de Flux.....	15
5.1. Introduction.....	15
5.2. Classification des Flux.....	16
6. Le Rôle du Temps.....	17
6.1. Sur la Nature des Contraintes de Synchronisation Multimédia.....	17
6.2. Les Unités de Synchronisation.....	18
6.3. De l'intérêt du Concept d'Unité de Synchronisation.....	20
6.4. Vous avez dit Temps-Réel ?.....	21
7. La Synchronisation Intra-Flux.....	24
8. La Synchronisation Inter-Flux.....	25
9. La Synchronisation Inter-Sites.....	26
10. L'Asynchronisme des Traitements.....	27
10.1. L'Asynchronisme du Système de Communication.....	28
10.2. L'Asynchronisme des Horloges.....	31
10.3. L'Asynchronisme du Système d'Exploitation.....	31
10.4. L'Asynchronisme du Matériel.....	34
11. La Prise en Compte de l'Asynchronisme.....	34
11.1. La Notion d'Intervalle de Validité Temporelle.....	34
11.2. La Notion de Gigue.....	35
11.3. La Notion de Dérive.....	36
12. Conclusion.....	38
Chapitre 3 : Modélisation Formelle de la Synchronisation Multimédia.....	39
1. Introduction.....	39
2. Organisation du Chapitre.....	41
3. Que Modéliser ?.....	41
4. Comment Modéliser ?.....	42
5. Choix d'un Modèle Formel.....	44
6. Les Réseaux de Petri et la Synchronisation Multimédia.....	47
6.1. Méthodologie de Comparaison.....	47
6.2. Les Réseaux de Petri Temporisés.....	48
6.3. Les Réseaux de Petri Temporels.....	53
6.4. Les Réseaux de Petri à Arcs Temporels.....	57
7. Conclusion.....	61
Chapitre 4 : Les Réseaux de Petri à Flux Temporels.....	63
1. Introduction.....	63
2. Organisation du Chapitre.....	63
3. Vers une Nouvelle Sémantique de la Synchronisation.....	64
3.1 Points de Vue.....	64
3.2 Synchronisation et Unités de Synchronisation.....	64

3.3 Instants Fondamentaux.....	65
3.4 Formalisation de la Problématique en Environnement Faiblement Synchron.....	66
3.5 Vers une sémantique de la Synchronisation en Environnement Faiblement Synchron.....	72
4. Définition Formelle des RdPFT.....	79
5. Sémantique Formelle des RdPFT.....	80
5.1. Etat d'un RdPFT.....	81
5.2. Règles d'Evolution entre Etats.....	81
5.3. Règles de Tir des RdPFT.....	82
6. Propriétés des RdPFT.....	92
7. Exemples.....	102
7.1.Un Schéma de « Lip-Synchronisation ».....	102
7.2.Une Application Multimédia.....	104
8. Conclusion.....	109
Chapitre 5 : Les Réseaux de Petri Structurés à Flux Temporels.....	115
1. Introduction.....	115
2. Organisation du Chapitre.....	116
3. Les Réseaux de Petri Structurés à Flux Temporels.....	116
4. Structures de Synchronisation et RdPSFT.....	117
4.1. Structures de Synchronisation Elémentaires.....	117
4.2. Structures Temporelles Composées.....	119
5. Réduction et Vérification des RdPSFT.....	128
5.1. Réduction de Faisceaux Elémentaires.....	128
5.2. Réduction de Faisceaux et de RdPSFT.....	129
5.3. Exemple.....	134
5.4. Vérification Temporelle des Schémas de Synchronisation.....	136
6. Conclusion.....	138
Chapitre 6 : Les Réseaux de Petri Hiérarchisés à Flux Temporels.....	139
1. Introduction.....	139
2. Organisation du Chapitre.....	140
3. Introduction à l'Hypermédia.....	140
3.1. Définition de la Notion de Système Hypermédia.....	141
3.2. Les Problèmes Soulevés par l'Hypermédia.....	141
4. Panorama des Modèles Existants.....	143
4.1. Le Modèle de Référence « Hypertexte » Dexter.....	143
4.2. Le Modèle Hypermédia d'Amsterdam.....	145
4.3. Les Apports des Réseaux de Petri pour la Modélisation Hypermédia.....	146
4.4. Pourquoi un Nouveau Modèle ?.....	148
5. Modélisation des Composants Atomiques.....	150
6. Modélisation des Composants Composites.....	151
7. Modélisation des Liens.....	152
7.1. Liens Temporels.....	152
7.2. Liens Dynamiques.....	154
7.3. Liens Synchrones et Asynchrones.....	157
7.4. Modélisation de l'Indéterminisme.....	158
8. Vision Globale de la Modélisation des Systèmes Hypermédias.....	158
9. Les Réseaux de Petri Hiérarchisés à Flux Temporels.....	160
9.1. Définition.....	160
9.2. Règles de Tir.....	161
9.3. Etat d'un RdPHFT.....	162
9.4. Modélisation et Simulation des Evénements Asynchrones.....	163
10. Exemples.....	165
10.1. Une Solution à la « Lip-synchronization » à l'Aide des RdPHFT.....	165
10.2. Une Application Hypermédia.....	167
11. Conclusion.....	169

Chapitre 7 : Conclusions et Perspectives.....	171
1. Méthodologie.....	173
2. Outils.....	175
3. Réalisations.....	176
4. Perspectives.....	177
 Annexe : Notations.....	 181
Bibliographie de l'Auteur.....	183
Bibliographie	187



Chapitre 1

Introduction

L'engouement suscité dans les sociétés « technologiquement évoluées » par les systèmes « multimédias » tient certainement en partie aux structures mentales et mythes fondateurs qui sous-tendent les artifices technologiques liés à cette évolution « naturelle » des systèmes informatiques. Ainsi, en se plaçant dans une perspective ontologique, la « société multimédia » apparaît immanente à l'histoire et à l'évolution de l'homme. Peintures rupestres de Lascaux, invention du langage, de l'écriture, invention de l'imprimerie, fresques de la chapelle Sixtine, invention du téléphone, du cinéma, de la télévision, du micro-ordinateur, autant de grandes étapes artistiques, culturelles et scientifiques participant de la même sémiotique de la communication et qui se trouvent soudainement condensées dans les systèmes multimédias.

L'écran informatique renvoie bien trop souvent encore une image alphanumérique qui est un pâle reflet de la « réalité ». L'incapacité qu'a l'utilisateur à percevoir à travers le miroir de l'écran une « image » conforme à sa représentation sensible du monde le confronte à une frustration narcissique et l'oblige à combler de façon contre-productive le fossé sémantique auquel il est ainsi exposé. Ainsi il est temps que les systèmes informatiques reproduisent dans un cadre numérique les paradigmes primitifs de la communication humaine. Le passage d'un « alpha » numérique » à un « oméga » numérique où l'homme-pygmalion aura face à lui une machine à son image se réalise progressivement à partir d'une évolution anthropomorphique des systèmes informatiques. Les systèmes multimédias introduisent une étape majeure de ce processus évolutionniste, celle où les systèmes informatiques sont capables de traiter uniformément sous une forme numérique la perception sensible et immédiate qu'a l'homme de son environnement.

En supprimant les frontières spatiales, temporelles et culturelles le multimédia emprunte également aux mythes du « don d'ubiquité » et « de la tour de Babel ». En effet les performances et la densité des réseaux à hauts-débits offrant des possibilités de communication « temps-réel » d'informations multimédias à une échelle planétaire, sont en train de donner une réalité virtuelle (i.e. sous une forme numérique) au mythe du village global, évoqué par Mac Luhan, et à la société teilhardienne où les choses et les êtres sont et valent par leur désir et leur capacité à s'unir [MLu67][Teil55]. Ce concept de village global est en train de se concrétiser à travers la notion de

système distribué hypermédia introduite par le World-Wide Web [VSW94]. En effet, en permettant de définir des relations sémantiques entre des objets multimédias, l'hypermédia apparaît comme une extension de la notion de multimédia fournissant un support au cheminement de la pensée. Ainsi, le concept d'hypermédia distribué en permettant des liens sémantiques entre des objets multimédias répartis préfigure une nouvelle forme émergente d'« esprit planétaire ».

On peut également expliquer et justifier l'irruption massive du multimédia dans la société contemporaine par des considérations structurales et économiques. En effet, d'après les travaux de François Flückiger de l'Université de Genève les êtres humains acquièrent 80 p. 100 de leurs connaissances par la vue, 11 p. 100 par l'ouïe, 3,5 p. 100 par l'odorat et 1,5 p. 100 par le toucher [Sti95]. De plus ces travaux ont mis en évidence que nous retenons 20 p. 100 de ce que nous voyons, 30 p. 100 de ce que nous entendons, et 50 p. 100 de ce que nous voyons et entendons en même temps. Ainsi les systèmes multimédias de part leurs capacités à présenter simultanément à l'utilisateur de l'audio, de la vidéo, du texte, des images, constituent a priori un moyen privilégié et performant pour accéder à la connaissance et pour gérer d'une façon efficace le nouvel étalon-or de nos sociétés industrielles que constitue l'information.

Mythes, théories de l'apprentissage, de la communication et intérêts économiques se conjuguent donc pour rendre incontournable la « société multimédia » qui devrait avoir de profondes influences sociales, culturelles et économiques sur notre civilisation.

D'un point de vue informatique, la définition consensuelle de système multimédia recouvre les systèmes permettant le traitement intégré et banalisé, à des fins de présentation, communication ou filtrage, de médias (i.e. de médias de présentation, communication, stockage..) statiques (i.e. indépendants du temps tels que texte et image) et dynamiques (i.e. dépendants du temps tels que audio et vidéo). Dans cette définition le concept de banalisation signifie que le système offre, quelque soit le type de média, les mêmes opérations génériques d'entrée/sortie, communication, et plus généralement d'allocation de ressources. De même la notion de gestion intégrée de médias recouvre la possibilité de composer, sous la forme d'un objet multimédia, plusieurs objets monomédias ou, récursivement, multimédias logiquement corrélés. En outre, dans son acception stricte un système multimédia doit offrir la possibilité d'accéder à tout nouvel objet multimédia, ainsi défini, à partir de la même interface générique que celle offerte pour les objets monomédias ou multimédias connus initialement du système. Il convient ici de souligner que la représentation sous forme digitale de l'ensemble des médias est à la base de tout système multimédia tel qu'il vient d'être défini.

Un système multimédia est qualifié de distribué lorsqu'il permet un accès transparent à des objets multimédias constitués de composants élémentaires répartis sur plusieurs sites serveurs interconnectés par l'intermédiaire d'un réseau local ou grande distance. La capacité des systèmes multimédias à reproduire les schémas traditionnels de communication qui, par essence, mettent en jeu des informations élaborées et conservées de façon répartie, fait que ces systèmes sont indissociables de la notion de répartition [WB94]. Il apparaît donc que les systèmes multimédias imposent un saut conceptuel aux systèmes informatiques actuels: celui du passage de systèmes gérant des informations statiques à des systèmes gérant des informations dynamiques sous une forme intégrée, banalisée et distribuée.

Un système multimédia offre donc à l'utilisateur la capacité de gérer un espace d'objets multimédias lui apparaissant comme autant d'îlots autonomes d'information sans structuration de plus haut niveau. Ainsi, dans un système multimédia, toute tentative de structuration de cet espace d'information et de navigation entre ces îlots informationnels doit être prise en charge explicitement par le niveau applicatif. Un système multimédia devient hypermédia lorsqu'il permet de mettre en œuvre des relations sémantiques ou liens entre des objets multimédias. De plus, les liens gérés par un système hypermédia définissent non seulement des relations sémantiques entre des objets multimédias mais fournissent également autant de possibilités de navigation, prises en charge implicitement par le système, entre des objets multimédias sources et destinations du lien (qui dans le cas général peuvent être distribués). Un système hypermédia permet donc de structurer l'espace d'information multimédia offert à l'utilisateur en un réseau informationnel, également appelé réseau hypermédia. Ainsi lorsque l'utilisateur d'un système hypermédia accède à un objet multimédia c'est tout une structure réticulaire chargée de sens qui lui apparaît de façon sous-jacente. Nous appellerons objets hypermédiés ces « meta-objets » multimédias gérés par un système hypermédia. De même que la notion d'objet permet de reproduire des paradigmes « naturels » de communication, la notion d'objet hypermédia ne constitue pas un artefact informatique mais représente « une métaphore valant pour toutes les sphères de la réalité où des significations sont en jeu » [Lév93]. Ainsi multimédia et hypermédia apparaissent comme deux notions intimement liées, en effet non seulement l'hypermédia constitue une extension de la notion de multimédia, mais aussi le multimédia accède à une structuration sémantique à l'aide de l'hypermédia.

Le concept de système multimédia/hypermédia nécessite de reconsidérer l'architecture logicielle et matérielle des systèmes informatiques actuels, à la lumière des nouvelles contraintes induites par les données et applications multimédias/hypermédias en matière de dynamique, d'intégration et de répartition. En particulier, les systèmes multimédias doivent gérer des objets composites associés de façon sous-jacente à des scénarios de présentation où les divers composants sont mis en scène comme autant d'acteurs. Le respect de ces scénarios de synchronisation apparaît comme fondamental pour la sémantique des objets multimédias. On notera qu'un système multimédia fait intervenir deux niveaux de contraintes temporelles: celles associées à la gestion des médias dynamiques, qui sont en général définies lors de la capture du média, et celles associées à des scénarios de synchronisation, qui sont définies artificiellement à partir de considérations applicatives.

Ainsi les contraintes de synchronisation exprimées par le scénario de présentation associé à un objet multimédia explicitent une qualité de service devant être offerte par le système multimédia afin de satisfaire les exigences applicatives. Cette notion de qualité de service est indissociable de la notion d'objet multimédia et constitue une des caractéristiques novatrices les plus fondamentales des systèmes multimédias/hypermédias.

On doit noter cependant que ces contraintes de respect de qualité de service apparaissent dans une première approche en contradiction avec les caractéristiques asynchrones des systèmes distribués universels sur lesquels sont appelées à s'exécuter les applications multimédias/hypermédias. En effet ces systèmes exposent les applications multimédias/hypermédias à des durées de traitement non déterministes ainsi qu'à des risques de défaillances susceptibles de remettre en cause la cohérence des scénarios de synchronisation associés aux objets multimédias/hypermédias. Ainsi, ces considérations placent la gestion des contraintes de synchronisation au cœur de la problématique de la conception des systèmes multimédias/hypermédias. Problématique d'autant plus

critique que, comme nous le verrons par la suite, aucune sémantique satisfaisante n'avait été définie jusqu'alors pour la notion de synchronisation dans un environnement asynchrone.

Bien que visant à résoudre à la fois les problèmes fondamentaux de la surcharge et de la carence d'information les systèmes hypermédias introduisent paradoxalement des artefacts induisant une perte du sens de l'orientation et une surcharge de connaissances lors du parcours d'un réseau hypermédia par un utilisateur [Conklin87]. L'accroissement permanent de la taille et de la complexité structurelle des applications hypermédias distribuées ne fait que renforcer ces effets indésirables et souligne par là même la nécessité de mettre en œuvre des méthodologies permettant de maîtriser la complexité et plus généralement la qualité des systèmes hypermédias.

De plus, le concept de lien hypermédia autorise l'utilisateur à interagir de façon asynchrone sur le déroulement d'un scénario de présentation multimédia afin d'orienter son déroulement ou de naviguer vers un autre noeud du réseau hypermédia. Cette possibilité rend nécessaire une clarification des façons dont les contraintes de synchronisation multimédias peuvent se combiner avec les événements asynchrones générés par l'utilisateur ou plus généralement l'environnement. De plus, si l'on perçoit bien ce que signifie attacher (ou « ancrer » selon la terminologie consacrée) un lien à une portion de texte cette notion apparaît beaucoup plus floue lorsqu'il s'agit d'un objet multimédia. Ainsi il apparaît nécessaire de clarifier la notion de synchronisation dans le contexte des systèmes hypermédias.

Les systèmes multimédias/hypermédias introduisent donc plusieurs problématiques scientifiques nouvelles qui appellent autant de contributions dans le domaine de la recherche informatique. Il importe cependant de constater que l'évolution technologique permet d'appréhender la disponibilité à court terme d'architectures matérielles offrant une qualité de service faisant de la conception de systèmes multimédias distribués une problématique essentiellement d'ordre logiciel. Le problème de la conception de systèmes multimédias n'en est pas pour autant résolu, car en mettant en œuvre simultanément des contraintes de synchronisation et de temps, ainsi que des contraintes de structuration logiques et sémantiques, les applications multimédias et hypermédias réunissent les problématiques les plus aiguës du génie logiciel. Ainsi, le saut conceptuel induit par le multimédia entraîne un accroissement de la complexité du logiciel qui souligne la nécessité d'utiliser des méthodes permettant d'assurer la qualité du logiciel multimédia. Aussi, conformément aux principes de base du génie logiciel prônant que « ce qui s'énonce clairement se conçoit bien », l'utilisation de méthodes permettant de spécifier de façon complète, aisée et précise des systèmes multimédias/hypermédias est un préalable incontournable à la conception de ces systèmes [BS96]. Les modèles ainsi obtenus permettent en particulier de dériver formellement les paramètres de qualité de service ainsi que les mécanismes devant être mis en œuvre dans les différents niveaux d'une architecture multimédia/hypermédia [PLL96].

La complexité du logiciel multimédia/hypermédia rend difficile voire impossible de garantir la qualité de cette nouvelle catégorie de logiciel à partir de méthodes de conception informelles ou empiriques. Par contre, en reposant sur des bases mathématiques, les techniques formelles offrent à l'auteur d'une application multimédia/hypermédia une sémantique formelle complète et dépourvue d'ambiguïté, ainsi que des techniques de simulation, de vérification et d'analyse des propriétés logiques et temporelles des programmes. Ainsi les méthodes formelles permettent de supprimer les syndromes auxquels sont exposés les programmes développés à partir de méthodes empiriques et informelles. En effet, sans l'adoption de méthodes formelles proposant des

structures temporelles, logiques et sémantiques, les applications multimédias et hypermédias sont exposées à des chaos (temporel, logique et sémantique) étendant et accentuant considérablement le syndrome du plat de spaghettis qui a terrassé le génie logiciel dans ses jeunes années.

Le travail réalisé dans le cadre de cette thèse s'inscrit dans la pragmatique de recherche du groupe de recherche Outils Logiciels pour la Communication (OLC) du Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS. Les thèmes de recherche traditionnellement abordés par le groupe OLC portent essentiellement sur la conception formelle de protocoles de communication. Les limitations des protocoles actuellement utilisés pour le transport d'information d'objet multimédia ont amené, dès 1991, le groupe OLC à conduire une réflexion sur les nouvelles exigences induites en matière protocolaire par les profils applicatifs multimédias et hypermédias. Ces travaux ont été essentiellement conduits dans le cadre du projet CESAME, un projet de recherche sur la conception formelle de systèmes distribués multimédias et coopératifs, financé par France-Télécom et associant le CNET, le CCETT et le CNRS [DP94]. Il est apparu que cette étude devait en tout premier lieu débiter par une définition claire et non ambiguë des caractéristiques fondamentales des applications multimédias et hypermédias. En effet la quasi totalité des travaux de recherche, réalisés jusqu'alors, abordaient la conception des systèmes multimédias selon un point de vue exclusivement technologique sans aucune considération méthodologique permettant d'avoir une compréhension claire et complète de ce nouveau champ de recherche. Ce constat, associé à l'expertise et à la longue pratique du groupe OLC dans le domaine des techniques de spécification formelles nous a conduit naturellement vers l'utilisation des méthodes formelles afin de décrire précisément les problématiques du multimédia et de l'hypermédia. Comme nous le verrons dans cette thèse cette approche s'est avérée concluante et doublement enrichissante. En effet, ces travaux ont permis d'une part de faire évoluer le champ général des techniques de spécification formelle et d'autre part d'obtenir des éclaircissements déterminants dans le domaine des systèmes multimédias et hypermédias. Enfin d'un point de vue global cette approche s'est avérée très concluante car elle a effectivement permis de dériver formellement des protocoles de communication adaptés au transport d'informations multimédia, à partir de la modélisation formelle des systèmes multimédias/hypermédias développée dans cette thèse [ACCCD94].

Ainsi, en synthétisant les considérations précédemment énoncées, il nous est apparu nécessaire, afin d'assurer la qualité des systèmes multimédias et hypermédias, de disposer de méthodes de spécifications formelles permettant de décrire de façon précise et complète les structures temporelles, logiques et sémantiques intrinsèques à ces systèmes. Tel est l'objectif de cette thèse qui, considérant les limitations des formalismes existants pour résoudre les nouvelles problématiques introduites par les systèmes multimédias/hypermédias, introduit un nouveau modèle, basé sur une extension temporelle des réseaux de Petri, permettant de spécifier dans un cadre formel, de façon aisée, précise et complète les structures temporelles, logiques et sémantiques inhérentes aux applications multimédias et hypermédias. Ce nouveau modèle se distingue des contributions ayant été réalisées jusqu'alors dans ce domaine car il fournit un cadre formel unificateur permettant de modéliser les structures fondamentales des systèmes multimédias/hypermédias tout en adoptant le cadre général des systèmes distribués synchrones ou faiblement synchrones. En particulier ce nouveau modèle fournit une sémantique formelle des notions fondamentales de synchronisation multimédia et hypermédia.

Ce nouveau modèle est progressivement introduit dans ce mémoire selon le plan suivant:

Le deuxième chapitre aborde et définit de façon informelle la problématique de la synchronisation multimédia. Suite à une étude des techniques conventionnelles de synchronisation multimédia et à l'analyse des limitations de ces approches, ce chapitre analyse de façon détaillée les structures de données et les structures de synchronisation mises en jeu dans un système multimédia distribué. La nature des contraintes temporelles induites par les traitements multimédias est ensuite étudiée. Soucieux de disposer de concepts les plus généraux possibles nous adoptons des hypothèses minimales devant être satisfaites par le système multimédia distribué. A ce titre on se place dans le cadre général d'un système distribué faiblement synchrone. Après avoir justifié le choix d'un tel cadre, nous exposons ses implications sur la problématique de la synchronisation multimédia.

Après avoir mis en évidence l'intérêt de disposer d'un modèle temporel permettant de spécifier les structures temporelles intrinsèques aux scénarios de présentation multimédias, le troisième chapitre met en évidence les propriétés devant être offertes par un tel modèle ainsi que les limitations inhérentes aux solutions proposées jusqu'à ce jour. Ce chapitre identifie dans un premier temps les traits fondamentaux qu'il est important de pouvoir modéliser dans un système multimédia. Partant de ces considérations, les caractéristiques générales d'un modèle temporel pour la spécification des contraintes de synchronisation multimédia sont mises en évidence. La confrontation de ce modèle général aux principales techniques de modélisation susceptibles d'être adoptées pour la modélisation des systèmes multimédias, désigne le formalisme des réseaux de Petri comme le formalisme le plus adapté. Une étude critique des réseaux de Petri étendus par le temps montre alors les limitations des formalismes proposés jusqu'à ce jour, et souligne la nécessité de proposer une nouvelle extension temporelle comblant les lacunes des modèles préexistants, tout en héritant de leurs acquis théoriques.

Un nouveau modèle permettant une spécification formelle complète et aisée des structures temporelles mises en jeu dans un système multimédia est alors introduit dans le quatrième chapitre. A partir d'une analyse détaillée de la problématique de la synchronisation dans les systèmes faiblement synchrones, ce chapitre introduit dans un premier temps une sémantique de la notion de synchronisation dans les systèmes faiblement synchrones. Une nouvelle extension temporelle aux réseaux de Petri, appelée Réseaux de Petri à Flux Temporels (RdPFT), permettant de modéliser simplement et complètement cette problématique est alors introduite, et sa sémantique formelle est explicitée. Puis, les propriétés fondamentales de ce nouveau modèle sont étudiées, et deux exemples de synthèse illustrent les possibilités offertes par ce nouveau formalisme.

Le cinquième chapitre pose les bases formelles permettant d'utiliser ce nouveau formalisme afin de modéliser la notion de composant logique d'un système multimédia/hypermédia. Après avoir introduit le modèle RdPSFT nous montrons en quoi ce modèle est bien adapté à la modélisation de schémas de synchronisation mettant en jeu des flux continus de traitements. En particulier nous montrons comment les structures de synchronisation informellement introduites dans le second chapitre trouvent une expression formelle à l'aide des RdPSFT. Nous montrons ensuite comment ces structures de synchronisation peuvent être réduites de façon à leur appliquer des techniques de vérifications temporelles. Puis cette technique de réduction-vérification sera étendue à des RdPSFT de complexité arbitraire. Les propriétés algébriques de cette technique de réduction seront alors étudiées. Ces techniques de réduction permettent d'introduire formellement la notion de structure logique d'un système multimédia. Finalement la dernière partie de ce chapitre est consacrée à l'étude d'un exemple.

Il est ensuite montré dans le sixième chapitre que cette nouvelle technique de spécification permet de modéliser dans un cadre formel unificateur les structures temporelles, logiques et sémantiques des systèmes hypermédias. La première partie de ce chapitre est consacrée à une brève introduction aux notions de système et de synchronisation hypermédias. Un panorama des techniques proposées pour la modélisation des systèmes hypermédias est ensuite exposé. Nous montrons en particulier la contribution du formalisme des réseaux de Petri dans le domaine de la modélisation des systèmes hypermédias. A partir des limitations des modèles existants et des contributions apportées par les Réseaux de Petri nous montrons alors comment le modèle RdPFT peut être étendu afin de permettre une modélisation aisée et complète des structures fondamentales d'un système hypermédia.

Finalement des considérations méthodologiques et des exemples de réalisations sont abordés en conclusion.



Chapitre 2

Problématique de la Synchronisation Multimédia

1. Introduction

La notion d'information multimédia est communément utilisée de façon informelle et intuitive. L'objet de ce chapitre est de caractériser de façon plus précise cette notion et de mettre en évidence la structure de données sous-jacente à la notion d'information multimédia. De plus, dans ce chapitre, on déterminera les propriétés fondamentales devant être satisfaites par un système multimédia pour garantir le respect de la sémantique des médias présentés à l'utilisateur.

Les opérations de présentation sont fondamentales dans un système multimédia. En effet la finalité d'un système multimédia est la présentation à un ou plusieurs utilisateurs de données multimédias totalement intégrées au système [WB94]. La validité d'un tel système repose donc sur la validité de la présentation des médias observés par le ou les utilisateurs. En particulier, les capacités de synchronisation temporelle, de représentation numérique et de synchronisation spatiale de plusieurs médias sont les trois concepts fondamentaux permettant d'assurer une gestion intégrée de l'ensemble des médias ainsi que le respect de la sémantique des opérations de présentation. Chacune de ces trois potentialités apparaît donc nécessaire à la mise en oeuvre d'une application multimédia. Partant de ces considérations et du fait que la répartition et donc la communication de l'information est sous-tendue à tout système multimédia, des efforts ont été et sont réalisés pour définir et normaliser chacune de ces potentialités. Les résultats les plus aboutis ont été obtenus pour les standards de représentation de l'information multimédia [Lu94] et pour la définition de structures logiques et spatiales de documents multimédias [BB91]. La maturité des travaux réalisés dans le cadre des structures logiques et des formats de représentation de l'information multimédia s'explique par le fait qu'ils s'inscrivent naturellement dans la continuité des travaux réalisés pour les structures de documents et de données traditionnelles (i.e. statiques). Par contre, les contraintes de synchronisation devant être satisfaites par les données multimédias introduisent un saut conceptuel dans les systèmes et applications informatiques: celui du passage d'un environnement statique à un environnement dynamique et temporellement contraint. Ainsi, la problématique de la synchronisation d'un système multimédia

apparaît à ce jour comme un nouveau problème, susceptible cependant de s'inspirer de travaux développés dans le cadre des systèmes distribués temps-réels [Fer94].

Les opérations de présentation de données multimédias peuvent être réalisées suite à des événements définis de façon statique (i.e. l'occurrence d'une date) ou dynamique (i.e. l'occurrence d'un événement asynchrone généré par l'environnement ou par l'utilisateur). Ainsi un système multimédia doit mettre en oeuvre des schémas de synchronisation. Les schémas de synchronisation définis de façon statique donnent lieu à des applications présentant à l'utilisateur des scénarios *immuables*. Ainsi, dans le contexte de la présentation ou plus généralement du traitement de données multimédias, on qualifiera de *synchronisation multimédia* tout schéma de synchronisation mettant en jeu des événements définis de façon *statique*. Les schémas de synchronisation multimédias sont à opposer aux schémas de synchronisation mettant en jeu des événements asynchrones dont la date d'occurrence ne peut être définie a priori et qui participent à ce que nous nommerons la *synchronisation hypermédia*.

Il convient de remarquer, pour souligner l'importance de la synchronisation multimédia, que la définition commune¹ du concept de système multimédia fait appel de façon sous-jacente à la notion de synchronisation multimédia. De plus, nous verrons ultérieurement que la synchronisation hypermédia s'appuie sur les services de la synchronisation multimédia. Il apparaît donc que la synchronisation multimédia est au coeur de la problématique de la conception des systèmes multimédias. Ainsi, ce chapitre est consacré exclusivement à l'étude des caractéristiques de la synchronisation multimédia.

2. Organisation du Chapitre

Suite à une étude des techniques conventionnelles de synchronisation multimédia et à l'analyse des limitations de ces approches, ce chapitre analyse de façon détaillée les structures de données et les structures de synchronisation mises en jeu dans un système multimédia distribué. Nous étudions ensuite la nature des contraintes temporelles induites par les traitements multimédias. Soucieux de disposer de concepts les plus généraux possibles nous adoptons des hypothèses minimales devant être satisfaites par un système multimédia distribué. A ce titre on se place dans le cadre général d'un système distribué faiblement synchrone. Après avoir justifié le choix d'un tel cadre, nous exposons ses implications sur la problématique de la synchronisation multimédia.

3. Le Paradigme de la Régie de Production

La spécification et la conception de tout système informatique doit être basée sur une interprétation d'un phénomène extrait du monde réel [Mey88]. En effet les archétypes de comportement empruntés au monde réel permettent de mettre en évidence des paradigmes qui par leur capacités d'abstraction, de modélisation et leur caractère éprouvé, s'avèrent être des puissants outils conceptuels lorsqu'on aborde une problématique. L'industrie du cinéma, grâce à son rôle précurseur, constitue indiscutablement la communauté la plus

¹Dans son acception commune, on entend par système multimédia tout système susceptible de présenter simultanément plusieurs médias à un utilisateur. La notion de simultanéité utilisée dans cette définition apparaît comme une vulgarisation de la notion de synchronisation multimédia.

expérimentée dans le domaine de la production multimédia. A ce titre, l'analyse des méthodes de travail des professionnels de la production cinématographique est la plus à même de permettre de dégager un paradigme naturel pour le traitement des flux multimédias. Ainsi, dans le domaine de l'industrie du cinéma, un paradigme éprouvé pour le traitement des flux multimédias peut être mis en évidence : il s'agit du *paradigme de la régie de production*. En effet les studios de production de films effectuent les montages cinématographiques à partir de régies de production permettant de réaliser des opérations génériques sur plusieurs pistes audio et vidéos².

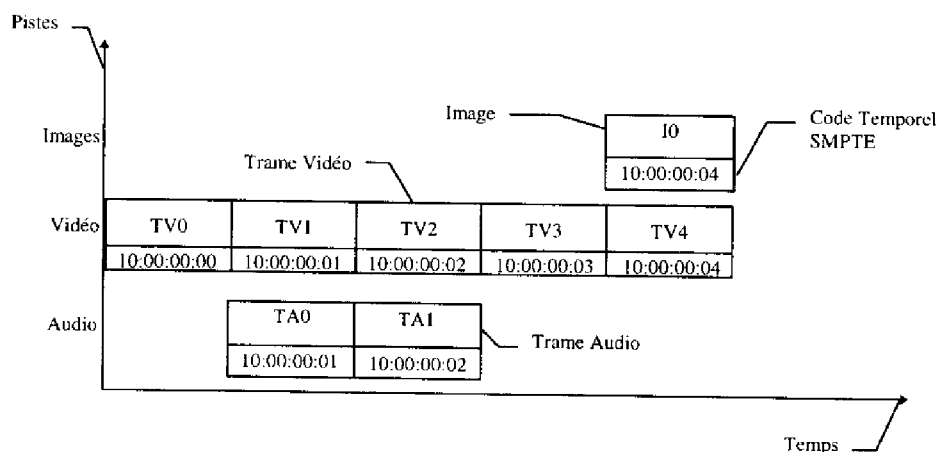


Figure 2.1 : Le Paradigme de la Régie de Production. Un scénario de synchronisation intégrant trois flux et spécifiant des contraintes de synchronisation inter et intra-médias à l'aide du paradigme de composition de la régie de production.

Les opérations communément réalisables à partir d'une régie de production sont les suivantes :

- l'association d'un code temporel à chaque unité d'information. Ces codes temporels sont standardisés par la SMPTE (Society of Motion Pictures and Television Engineers) et permettent d'associer un identificateur séquentiel codée sous la forme (heure,minute,seconde,image) à chaque trame d'une piste vidéo ou audio [SMPTE86]. En associant un instant absolu à chaque trame d'une piste audio ou vidéo les codes temporels SMPTE permettent de spécifier des scénarios de synchronisation intra et intermédias (Figure 2.1).
- le Contrôle d'Accès Temporel (CAT) aux trames d'informations [Lit94]. Ce contrôle est réalisé à partir d'opérations permettant l'accès direct à un échantillon audio ou vidéo (repéré par son code temporel), la recherche rapide en avant et en arrière d'une trame d'information, l'arrêt ou le démarrage de la présentation d'une piste audio ou vidéo.
- le mixage de flux qui permet de combiner plusieurs pistes issues de sources distinctes en une seule piste destinée à être consommée par un périphérique multimédia.

²Une piste est une séquence de trames d'informations audio ou vidéo.

- le filtrage, qui consiste à appliquer un traitement ou un effet spécial sur les trames d'informations d'une piste et qui permet ainsi d'en modifier le contenu.
- des opérations de type couper-coller sur trames d'une ou plusieurs pistes.

Le paradigme de la régie de production constitue une solide base de réflexion sur la problématique de la synchronisation dans les systèmes multimédias. En particulier, les codes temporels SMPTE permettent de spécifier simplement des contraintes de synchronisation inter et intra médias, et par là même des scénarios de synchronisation multimédias. Les codes temporels SMPTE induisent un temps virtuel qui doit être mis en correspondance avec le temps universel (i.e. une date UTC) lors de la présentation effective d'un scénario multimédia. Ainsi, lors de la présentation d'un scénario multimédia, chaque couple (trame, code temporel) est mis en correspondance avec un triplet (trame, date, durée) comme illustré sur la Figure 2.2. Chaque piste est ainsi virtuellement superposée à une *ligne temporelle* décrivant l'évolution d'un temps réel absolu ou relatif au début du scénario. Le paradigme de composition de la Régie de Production rejoint ainsi celui de la *ligne temporelle*, proposé en particulier dans Quicktime [App191] et Hytime [Hyt92], consistant à définir des scénarios de présentation de données multimédias en alignant des pistes sur un axe temporel.

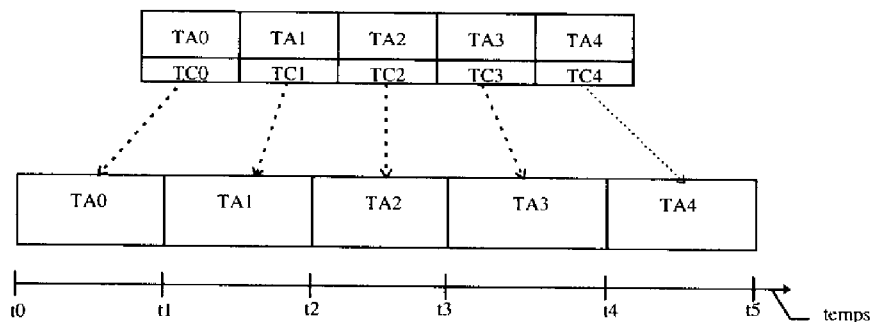


Figure 2.2 : Mise en correspondance des Codes Temporels avec le temps réel UTC. Lors de la présentation d'un média chaque trame est associée à une date et une durée de présentation

3.1 Les Limitations du Paradigme de la Régie de Production

L'association d'un code temporel à chaque trame d'information introduit les problèmes suivants :

- L'édition d'un média par l'intermédiaire d'opérations du type couper/coller est susceptible de ne plus garantir la cohérence en terme de séquentialité et de continuité des «codes temporels ». Cela tient au fait que les contraintes de synchronisation intra-media sont fortement couplées avec le média de représentation (i.e. un code temporel est attaché à chaque trame lors de la génération du média).
- Si on retire un média d'un scénario, des informations pertinentes pour la synchronisation des médias restants peuvent manquer. Il en est ainsi, par exemple, si on retire la piste vidéo du scénario représenté

dans la Figure 2.1, on ne possède alors plus assez d'informations temporelles pour assurer la synchronisation des deux pistes restantes. De plus, par définition, les codes temporels induisent une granularité de synchronisation fortement dépendante de la granularité du média de représentation vidéo.

- Ce paradigme de synchronisation d'informations multimédias a été conçu initialement dans un contexte « tout analogique » où les médias, sous forme analogique, sont gérés par des périphériques analogiques entièrement dédiés à leur présentation. Dans ce cadre, les problèmes de synchronisation sont résolus de façon ad hoc par des mécanismes « matériels » dédiés. Le passage au « tout numérique » permet un traitement « logiciel » des données multimédias et, par là même, leur intégration dans des systèmes informatiques non dédiés. Cette « banalisation » des données multimédias les expose à des *artefacts* issus du monde informatique. En particulier le strict déterminisme temporel sur lequel repose le paradigme de la régie de production et qui est assuré par une synchronisation réalisée à partir d'une horloge temps réel distribuée à l'ensemble des équipements du studio de production ou de diffusion est remis en cause dans le cadre général des systèmes multimédias distribués asynchrones.

Néanmoins, le paradigme de la régie de production constitue une solide base illustratrice de la notion de synchronisation multimédia. Ainsi, partant de ce paradigme et de ses limitations intrinsèques nous allons définir dans les sections suivantes un modèle conceptuel général pour la synchronisation multimédia.

4. Les Unités d'Information

Par définition, un système multimédia peut être amené à gérer de façon intégrée une grande diversité de médias de représentation. Chacun des médias de représentation est issu d'une source de type média de stockage, d'acquisition, ou programme et, suite éventuellement à des filtrages³ et communications, est présenté à l'utilisateur (respectivement stocké) à l'aide d'un média de présentation (respectivement sur un média de stockage). Ainsi la présentation d'une information multimédia à un ou plusieurs utilisateurs est obtenue à partir d'un « pipeline » de traitements élémentaires⁴ conformément à la Figure 2.3.

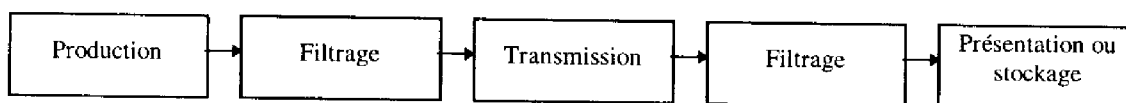


Figure 2.3 : Le « pipeline » des traitements multimédias. Un traitement multimédia apparaît comme une composition de traitements élémentaires concourant à un résultat global.

La structure fonctionnelle décrite par la Figure 2.3 soulève le problème de la nature des données, ici les médias de représentation, circulant dans le « pipeline ». Ces données sont créées suite à la discrétisation et au codage

³Un filtrage peut consister en une opération de compression ou décompression par exemple.

⁴Dans ce pipeline les opérations de filtrage et de transmission sont mises en jeu de façon optionnelle en fonction du profil applicatif.

d'informations multimédias dynamiques (la voix par exemple) ou statiques (un texte par exemple) lors de leur acquisition à partir d'un périphérique d'entrée. Cette discrétisation et ce codage induisent la notion d'échantillon d'information. Ces échantillons élémentaires représentent les particules constitutives des médias de représentation ; à ce titre ces échantillons élémentaires seront appelés *atomes d'information*. Ainsi, le média de représentation relatif à une information multimédia est constitué d'une séquence d'atomes d'information résultant d'un processus de codage de l'information. Par exemple, une vidéo est constituée d'une séquence de trames satisfaisant un certain standard de codage tels que MPEG1, MPEG2 ou H261. Les données circulant dans le « pipeline » précédemment évoqué sont donc le résultat d'opérations de filtrage, groupage/dégroupage, assemblage/désassemblage d'atomes d'information. Ainsi, plus généralement, on désigne par *unité d'information* toute entité regroupant un contenu informationnel ayant une sémantique pour un étape du « pipeline » ; c'est à dire toute information susceptible d'être acquise, filtrée, communiquée, présentée ou stockée. A une unité d'information correspond donc une suite d'atomes d'information. Par exemple, lors du transfert d'une voie audio sur une connexion de niveau transport plusieurs atomes d'information audio peuvent être adressés « en bloc » au service transport dans une même TSDU, dans ce cas pour le service de transport les unités d'information correspondent aux blocs d'atomes d'information audio qu'il traite comme un tout. De plus, chaque unité d'information est associée à un type de donnée multimédia de base tel que l'audio, le texte, la vidéo et cetera, conformément au type des atomes d'informations sous-jacents à l'unité d'information considérée.

La notion d'unité d'information est étroitement liée à celle de *granularité* de l'information. En effet, une unité d'information peut avoir un contenu informationnel plus ou moins structuré et abstrait. Ainsi pour un média de type vidéo une unité d'information peut contenir une trame, un plan, une séquence ou un film tout entier. De même pour un média de type audio une unité d'information peut être un phonème, un mot ou une phrase. On remarquera donc que la notion d'unité d'information ne correspond pas nécessairement à la plus petite entité informationnelle ayant une sémantique applicative. La granularité des unités d'informations est définie à partir de considérations applicatives ou de contraintes issues de l'environnement. A titre d'exemple l'unité d'information sur laquelle opère une application de vidéo à la demande peut être une image ou un film entier.

4.1 De l'Intérêt du Concept d'Unité d'Information

Comme illustré par la Figure 2.4 le concept d'unité d'information introduit une indirection entre les médias de représentation et les applications. Ainsi une ou plusieurs applications peuvent adopter des points de vue différents sur un même média de représentation et induire une « sémantique » applicative sur un média. Cette notion de point de vue sur les médias de représentation s'avère être un concept de base dans les systèmes multimédias. En effet, ces systèmes sont souvent conçus de façon hiérarchisée [Buf94], chaque niveau de la hiérarchie percevant les médias de représentation avec une certaine granularité, en général plus grossière au sommet de la hiérarchie qu'à la base. Ainsi, chaque niveau de la hiérarchie d'un système multimédia définit ses propres unités d'informations indépendamment de la granularité élémentaire des atomes d'informations.

Les unités d'information facilitent grandement la modification et l'édition des médias de présentation. En effet, comme illustré par la Figure 2.4 l'ordre des atomes d'information au sein d'un média de représentation peut être aisément changé sans induire de modifications au niveau de la structure du média de représentation. On peut

notamment remarquer que les atomes d'information sont un cas particulier d'unités d'information. Ainsi, en dissociant le média de représentation de son utilisation, la notion d'unité d'information supprime une des principales limitations du paradigme de la régie de production à savoir la nécessité de régénérer à partir d'une copie « maître » tout ou partie d'un média de présentation lors d'une opération de montage.

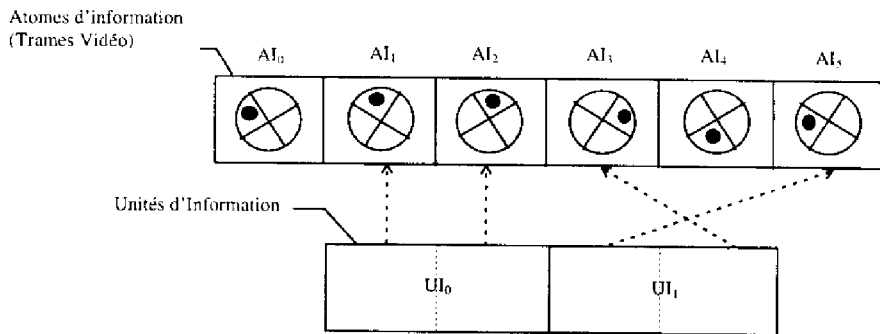


Figure 2.4 : La correspondance entre Particules d'Information et Unités d'Information.
 Les unités d'information introduisent un niveau d'indirection entre l'application et la structure du média de présentation défini par une suite d'atomes d'information. Chaque unité d'information UI , est associée à une suite d'atomes d'information (AI).

L'indépendance entre le média de représentation et l'information manipulée au niveau applicatif a été exploitée dans des systèmes multimédias expérimentaux ou commerciaux. Ainsi le concept d'unité d'information peut être rapproché de la notion de « track-segment » utilisée dans les fichiers « Movies » Quicktime [Way91] [DB93]. De même dans [RVR94] le concept de « primary bloc » s'apparente à la notion d'unité d'information.

5. La Notion de Flux

5.1 Introduction

La section précédente a mis en évidence qu'un média de représentation est structuré au niveau le plus élémentaire sous la forme d'une suite d'atomes d'information. La notion d'unité d'information a été ensuite introduite afin de permettre d'adopter des points de vue à granularité moins fine sur cette structure élémentaire. Les unités d'information permettent en particulier de mettre en correspondance la structure élémentaire d'un média de représentation avec des niveaux d'abstractions spécifiques des étapes de stockage, communication, filtrage et présentation. Pour chacune des étapes élémentaires du « pipeline » des traitements multimédias, un média de représentation apparaît sous la forme d'une suite d'unités d'informations. Ainsi, l'information de base manipulée par une application multimédia, à savoir la notion de média de représentation, peut être modélisée de façon unifiée sous la forme d'une suite d'unités d'informations. On appellera *flux* une telle structure de données.

Définition 2.1 : Une suite d'unités d'information $f = (u_i)_{i \in I}$ définit une structure de donnée appelée *flux d'unités d'information*.

La notion de flux est une notion fondamentale car unificatrice. En effet, le concept de flux peut être exploité à des fins d'acquisition, de stockage, de communication, de filtrage, ou de restitution. Ainsi une application multimédia peut être définie comme une application opérant un traitement sur plusieurs flux d'informations. La notion de flux apparaît donc comme une extension du concept de média de représentation.

5.2 Classification des Flux

Une classification des flux, en fonction de leurs propriétés temporelles, permet de dégager les mécanismes fondamentaux devant être mis en oeuvre par un système multimédia pour réaliser leur traitement et garantir le respect de leur sémantique temporelle.

Ainsi, les flux peuvent être classés en deux grandes familles :

- Les *flux statiques*, ou flux dont la sémantique de présentation est indépendante d'attributs temporels intrinsèques. On peut regrouper dans cette famille les médias tels que le texte, les images fixes et les images de synthèse. Un flux statique peut donc être modélisé sous la forme d'une suite, éventuellement réduite à un seul élément, *d'unités d'informations* non temporellement corrélées et dépourvues d'attributs temporels intrinsèques. Il est à noter qu'un flux statique ne se réduit pas nécessairement à une seule unité d'information; ainsi, par exemple, une suite d'images fixes ou de pages de texte constitue un flux statique.
- Les *flux dynamiques*, pour lesquels la sémantique de présentation est dépendante d'attributs temporels intrinsèques. Les médias tels que l'audio, la vidéo, les animations de synthèse appartiennent à cette famille. En effet, par exemple, la compréhension d'un flux audio exige que la fréquence de restitution des unités d'information le constituant soit égale à leur fréquence d'acquisition. Les flux dynamiques sont structurés sous la forme de *suites ordonnées d'unités d'informations*. Par exemple, une vidéo est une suite de trames vidéos, un message audio est une suite d'échantillons audios. Ainsi les flux dynamiques peuvent être modélisés sous la forme d'une suite d'unités d'informations temporellement corrélées et dotés d'attributs temporels intrinsèques. Le respect de la sémantique des flux dynamiques exige que la présentation de chaque unité d'information le composant satisfasse des contraintes temporelles en terme de date et de durée de présentation. La présentation à l'utilisateur d'un média dynamique dans un système multimédia distribué impose donc des contraintes temporelles sur les dates et les durées de réalisation de l'ensemble des traitements de filtrage, communication et présentation composant ce traitement réparti. Dans la classe des flux dynamiques on distingue les flux *continus* tel que la vidéo ou l'audio, c'est à dire les flux dont les unités d'information doivent être présentées de façon contiguë dans le temps. Les flux continus sont à opposer aux *flux discrets* c'est à dire composés d'unités de synchronisation non temporellement contiguës. La périodicité des dates de traitement des unités d'information permet également de distinguer des catégories de flux dynamiques telles que les flux *apériodiques*, *périodiques* ou *semi-périodiques*.

Un flux statique apparaît comme un cas particulier de flux dynamique dont les unités d'information ont une date et une durée de présentation indéfinies. De plus, un flux sera dit *homogène* ou *monomédia* lorsque toutes les

unités d'information le constituant ont le même type de donnée et sont donc associées au même média de présentation; dans le cas contraire le flux sera dit *hétérogène*. Outre le caractère dynamique ou statique des flux, d'autres caractéristiques permettent leur classification et leur description. En particulier la durée de vie d'un flux est un critère important impliquant la possibilité pour une application de réutiliser ou non ce flux. Ainsi, on désignera par flux *persistant* tout flux mémorisé dans une base de donnée. Parmi les flux non persistants, également qualifiés d'*évanescents*, on distingue la classe des flux « *vivants* » qui sont créés en temps-réel à partir d'un périphérique multimédia tel qu'une caméra ou un microphone par exemple. Les principaux attributs temporels des flux sont résumés par le tableau 2.1.

Type de Flux	Caractéristiques
Flux Statique	Sémantique indépendante du temps
Flux Dynamique	Sémantique liée au temps
Flux Continu	Présentation contiguë des unités d'information
Flux discret	Présentation non contiguë des unités d'information
Flux persistant	Mémorisé dans une base de données
Flux non persistant	Créé et supprimé dynamiquement
Flux évanescent	Présenté pendant une courte durée
Flux vivant	Créé en « temps-réel »

Tableau 2.1 : Les attributs temporels des flux

6. Le Rôle du Temps

6.1 Sur la Nature des Contraintes de Synchronisation Multimédia

Le tableau 2.1 met en évidence le caractère fondamental joué par les contraintes de synchronisation⁵ dans une application multimédia. En effet, il apparaît que les flux gérés par une application multimédia doivent satisfaire, de façon non nécessairement exclusive, deux types de contraintes de synchronisation, à savoir :

- des contraintes de synchronisation *naturelles*, c'est à dire intrinsèques aux flux. Le respect de ces contraintes de synchronisation lors de la présentation d'un flux à l'utilisateur est fondamental car il conditionne la bonne compréhension par l'utilisateur de l'information associée. Ces contraintes de synchronisation naturelles sont généralement définies lors de l'acquisition du flux et induisent une date et une durée de traitement pour chaque unité de synchronisation. En particulier, les flux *continus*, c'est à dire constitués d'une suite d'unités de synchronisation temporellement contiguës tels que les flux audio ou vidéo induisent des contraintes de synchronisation naturelles. C'est ainsi, par exemple, que les unités d'information (i.e. les trames) d'un flux vidéo respectant la norme SECAM doivent être présentées nominalement à l'utilisateur avec une fréquence de 1/25 s. Ces contraintes de synchronisation sont

⁵Par synchronisation on entend la gestion coordonnée d'événements dans le temps.

généralement assurées par les codecs de flux continus et à ce titre sont rarement spécifiées au niveau d'un langage d'auteur d'applications multimédias.

- des contraintes de synchronisation *synthétiques*. Ces contraintes de synchronisation sont imposées par le contexte applicatif et résultent d'une synchronisation abstraite spécifiée par l'auteur de l'application multimédia. Ainsi, par exemple, l'auteur d'une application multimédia peut spécifier que la présentation d'un flux doit commencer suite à l'occurrence d'un événement portant sur le contexte applicatif (occurrence d'une date ou d'une interaction utilisateur par exemple). En particulier, la présentation d'un flux statique à un utilisateur rend nécessaire la spécification et la mise en oeuvre par le système multimédia de contraintes de synchronisation synthétiques. En effet, pour présenter un flux statique à un utilisateur il est nécessaire de spécifier les dates d'activation et les durées des unités de synchronisation réalisant la présentation des unités d'information de ce flux. Il est à noter que des contraintes de synchronisation synthétiques peuvent se substituer à des contraintes de synchronisation naturelles. Il en est ainsi, par exemple, lorsqu'un flux vidéo est accéléré ou ralenti suite à une interaction utilisateur.

6.2 Les Unités de Synchronisation

Le traitement d'un flux multimédia à des fins de stockage, communication, filtrage ou présentation consiste à appliquer le traitement correspondant sur chacune des unités d'information composant ce flux. On appellera *unité de synchronisation* tout traitement réalisé sur une suite d'unités d'informations.

Définition 2.2 : Une *unité de synchronisation*, notée s , associée à un flux d'unités d'informations $f = (u_i)_{i \in I}$ est une tâche réalisant un traitement sur une suite d'unités d'information $(u_j)_{j \in J} \subset f$ extraite du flux f .

Ainsi, à tout flux d'unités d'informations correspond, lors de chaque étape du « pipeline », une suite ou flux d'unités de synchronisation. Le terme flux désignera donc, en fonction du contexte, une structure de données ou une structure de synchronisation.

Définition 2.3 : Une suite $g = (s_i)_{i \in I}$ d'unités de synchronisation définit une structure de synchronisation appelée *flux d'unités de synchronisation*.

La finalité d'une application multimédia est la présentation de flux à l'utilisateur. Ainsi, la perception temporelle de l'évolution d'une application multimédia est réalisée par l'utilisateur à travers les dates et durées de présentation des unités d'information associées aux flux gérés par l'application. Ces présentations, réalisées suite à des événements statiques ou dynamiques, sont mises en oeuvre en associant à chaque unité de synchronisation participant à la présentation considérée une date d'ordonnement et une durée de présentation. Etant donnée la structure en pipeline de la chaîne des traitements aboutissant à la présentation d'un flux multimédia, il apparaît que la date d'ordonnement associée à une unité de synchronisation réalisant une présentation induit des contraintes d'ordonnement pour les unités de synchronisation des étapes amont du « pipeline ». Ainsi, si nous considérons un pipeline de traitements $(T_i)_{i \in \{1, n\}}$ (avec T_n le traitement réalisant la présentation) tel qu'à tout traitement T_k est associé un délai de réalisation t_k , et si la date nominale d'ordonnement d'une unité de

synchronisation $US_{i,n}$ (i.e. la $i^{ème}$ unité de synchronisation traitée lors de l'étape n) participant à la présentation du flux est $d_{i,n}$, alors la date nominale d'ordonnancement au plus tard de l'unité de synchronisation correspondante $US_{i,k}$ pour l'étape T_k est donnée par⁶ :

$$d_{i,k} = d_{i,n} - \sum_{n=k}^{n-1} t_n$$

Les flux vivants n'offrent pas de grande latitude en matière d'ordonnancement car les dates d'ordonnancement des unités de synchronisation relatives à ces flux sont imposées par le traitement initial du pipeline qui produit en « temps-réel » le flux. En particulier les délais cumulés lors de la traversée du pipeline des traitements doivent être compatibles avec une utilisation interactive. Les délais maximum admissibles avec un contexte interactif sont donnés, en fonction de la nature du média, dans le tableau 2.2 [HSS92]. Par contre, l'ordonnancement des traitements relatifs à des flux persistants se prête à des mécanismes d'anticipation permettant de compenser la limitation éventuelle des ressources des divers composants du pipeline [LG90c]. Les unités de synchronisation constituent donc les entités ordonnancées par un système multimédia. Ainsi, la notion d'unité de synchronisation s'apparente à la notion de tâche dans un système multitâche.

Médium	Délai maximum en secondes
Voix	0.25
Vidéo (Qualité TV)	0.25
Vidéo Compressée	0.25
Texte	1
Données (transfert de fichier)	1
Image	1

Tableau 2.2: Les Délais maximum compatibles avec un contexte interactif.

Le traitement d'un flux induit donc l'ordonnancement séquentiel des unités de synchronisation réalisant le traitement considéré (i.e. stockage, communication, filtrage, présentation) sur chacune des unités d'information composant ce flux. Cependant, la relation d'ordre strict existant entre les dates d'ordonnancement des unités de synchronisation d'un flux est indépendante de l'ordre des unités d'information dans le flux (Figure 2.5). En effet, si chaque unité de synchronisation s_k d'un flux $g = (s_j)_{j \in J}$ est associée à une suite d'unités d'information

$(u_i)_{i \in I_k}$ (avec $I_k \subset I$) extraite d'un flux $f = (u_i)_{i \in I}$, alors le flux f « devient » $f' = \left((u_i)_{i \in I_i} \right)_{i \in J}$ lorsqu'on lui

applique g . En particulier les opérations de Contrôle d'Accès Temporel (CAT) précédemment évoquées sont susceptibles d'induire le bouleversement de l'ordre de traitement des unités d'information. Ainsi certaines unités

⁶On adopte ici l'hypothèse simplificatrice suivante: la granularité des unités de synchronisation est la même pour tous les traitements du « pipeline ».

d'informations peuvent être sautées (cas d'une avance accélérée), dupliquées (cas d'un ralenti), ou bien les unités d'information peuvent être traitées dans l'ordre inverse de l'ordre naturel du flux (cas d'une présentation « en marche arrière »). De même des CAT plus évolués peuvent réaliser virtuellement (*i.e.* sans modifier le flux d'unités d'informations) sur un flux d'unités d'information des opérations de type « couper/coller » ou « copier/coller », et induire ainsi un ordre de traitement totalement indépendant de l'ordre de la structure de donnée sous-jacente.

De même que la notion d'unité de synchronisation est intimement liée au concept de granularité de l'information, le concept d'unité de synchronisation induit celui de *granularité de synchronisation*. En effet, le traitement de plusieurs unités d'information peut être encapsulé dans une même unité de synchronisation et ainsi être ordonné de façon monolithique. A chaque unité de synchronisation correspond donc une suite d'unités d'informations, la longueur de cette suite définissant la granularité de synchronisation associée à l'unité de synchronisation considérée.

6.3 De l'Intérêt du Concept d'Unité de Synchronisation

Les unités de synchronisation sont à la synchronisation ce que les unités d'information sont à l'information: elles permettent de mettre en oeuvre sur le même flux divers schémas de synchronisation à granularité plus ou moins fine. Chaque niveau hiérarchique d'un système multimédia peut ainsi adopter sur un même flux une granularité de synchronisation qui lui est propre.

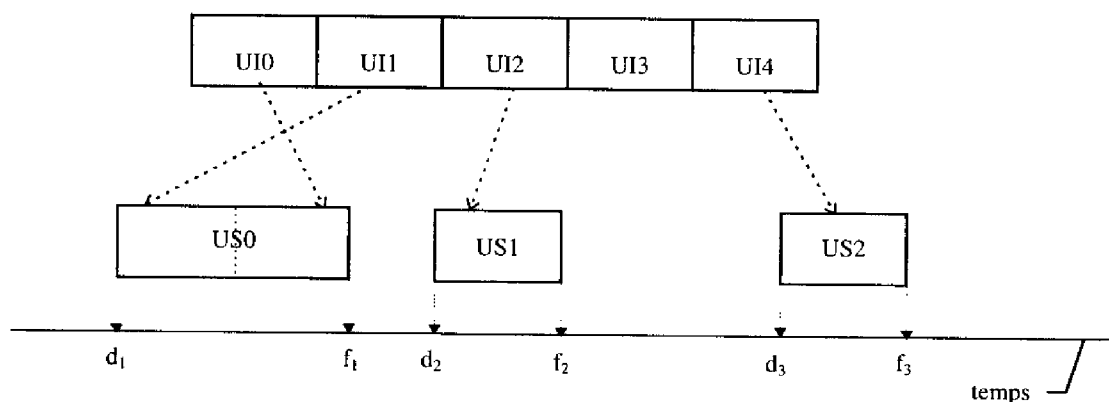


Figure 2.5 : Correspondance entre Unités d'Information et Unités de Synchronisation. Chaque unité de synchronisation, notée US_i , d'un flux est associée à une suite d'unités d'information (flèches en trait épais) ainsi qu'à une date d'ordonnement et de terminaison respectivement notées d_i et f_i (flèches en trait fin).

Le concept d'unité de synchronisation permet de moduler les schémas de synchronisation en fonction des ressources disponibles sur le système multimédia. L'utilisation de schémas de synchronisation de granularité minimale, c'est à dire associant une unité de synchronisation à chaque unité d'information, offre un contrôle optimal sur le respect des contraintes de synchronisation qui doivent être satisfaites au niveau du traitement d'un

flux. Cependant, il peut s'avérer très coûteux de mettre en oeuvre une granularité aussi fine en particulier pour les flux continus. Aussi en permettant de regrouper le traitement de plusieurs unités d'information sous une seule unité de synchronisation, les ressources nécessaires à la mise en oeuvre de ce traitement peuvent être réduites par le biais de factorisations d'opérations élémentaires. En particulier, le niveau de granularité influence directement le degré de sollicitation de l'ordonnanceur; de plus, les débits effectifs des contrôleurs d'entrées/sorties et les performances des protocoles dépendent des niveaux de granularité mis en jeu [NW94][Ste94].

La variabilité des schémas de synchronisation induite par le concept d'unité de synchronisation permet d'adapter le traitement d'un flux aux ressources disponibles. Cette caractéristique est susceptible d'assurer une certaine portabilité aux schémas de synchronisation ainsi spécifiés ainsi qu'une adaptabilité aux variations de disponibilité des ressources. Ainsi le traitement d'un flux peut-il être spécifié selon des schémas de synchronisation plus ou moins fins en fonction des ressources du système sur lequel doit être réalisé ce traitement. En particulier le concept d'unité de synchronisation permet, à la différence des « codes temporels » utilisés dans les régies de production, de s'affranchir de la granularité de synchronisation définie lors de l'acquisition du média.

6.4 Vous avez dit Temps-Réel ?

Par les schémas de synchronisation qu'elle met en oeuvre, une application multimédia s'apparente à une application multitâche distribuée. Cependant, dans le cas d'une application multimédia la mise en oeuvre des schémas de synchronisation inter-tâches n'a pas pour objectif la minimisation des coûts de traitement des tâches mais la satisfaction de contraintes de présentation ou plus généralement de contraintes de traitement associées aux flux. Ainsi, les contraintes de synchronisation naturelles ou synthétiques portant sur les unités de synchronisation induisent, outre une synchronisation logique entre les unités de synchronisation, la nécessité de mettre en oeuvre une synchronisation basée sur le *temps réel* perçu par l'utilisateur. En effet nous avons vu que les contraintes de synchronisation naturelles et synthétiques induisent une *date* de début d'exécution et une *durée* pour chaque unité de synchronisation. Ainsi une application multimédia induit un *ordonnement temps réel* de tâches ou unités de synchronisation. De façon à bénéficier des acquis réalisés dans le domaine du « temps réel » il paraît dès lors intéressant d'essayer de rattacher les unités de synchronisation multimédias à une des trois classes de tâches temps réel qui ont été identifiées jusqu'alors, à savoir: les tâches temps réel dur, ferme et mou [SR94]. Les profils temporels de ces trois classes d'application « temps réel » peuvent être définis par une fonction associant un bénéfice ou gain à la date de terminaison d'une tâche.

Par définition, si une tâche « temps réel dur » se termine avant une date de terminaison au plus tard on obtient un gain maximal, par contre si celle-ci se termine après la date de terminaison au plus tard on perd tout (Figure 2.6-a). A l'identique d'une tâche « temps-réel dur » une tâche « temps-réel ferme » possède une date de terminaison au plus tard, cependant le dépassement de cette date est moins critique et induit seulement un gain nul (Figure 2.6-b). Les tâches « temps réel mou » offrent quant à elles un gain décroissant vers zéro au fur et à mesure que la date de terminaison au plus tard est dépassée (Figure 2.6-c). Les unités de synchronisation multimédias peuvent se satisfaire d'une certaine flexibilité en matière de dates d'ordonnement et de terminaison. Cette flexibilité est induite par la perception subjective et imprécise du temps appréhendé par l'utilisateur [SE93]. Une unité de synchronisation multimédia s'apparente donc à la famille des applications « temps-réel mou ». Cependant à la

différence d'une application temps réel (i.e. temps-réel « dur », « ferme » ou « mou ») pour laquelle la date de terminaison de la tâche peut être quelconque à partir du moment où elle est inférieure à la date de terminaison au plus tard, une unité de synchronisation multimédia doit avoir une certaine persistance temporelle, c'est à dire qu'elle doit présenter des unités d'information pendant une durée de temps garantissant la bonne perception de l'information par l'utilisateur. Ainsi, à la différence des trois classes précédemment décrites le profil temporel d'une tâche multimédia introduit non seulement la notion de date de terminaison au plus tard mais aussi celle de date de terminaison au plus tôt.

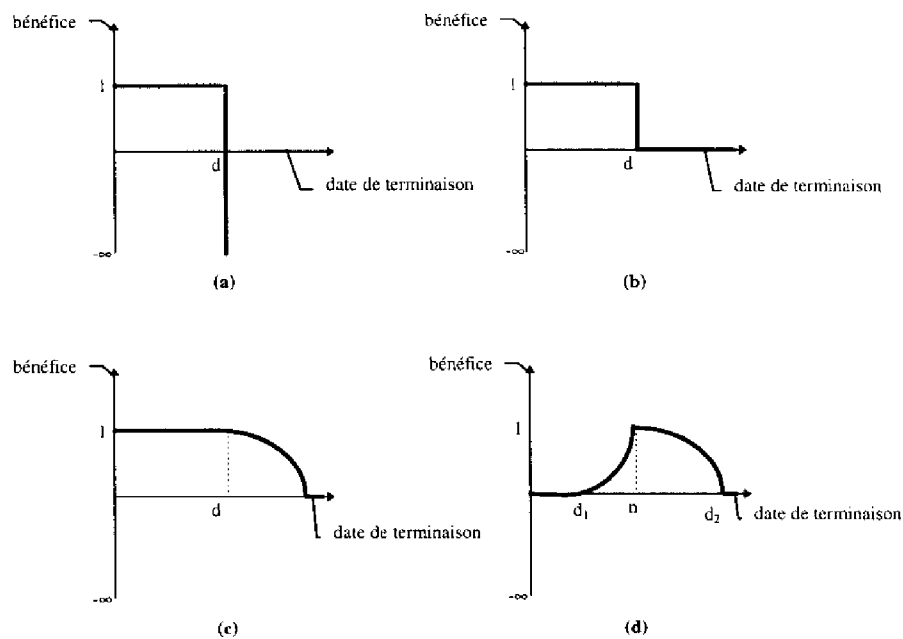


Figure 2.6 : Profils temporels des tâches « temps-réel ». (a) une Unité de Synchronisation « temps-réel dur » ; (b) une Unité de Synchronisation « temps-réel ferme » ; (c) une Unité de Synchronisation « temps-réel mou » ; (d) une Unité de Synchronisation « multimédia ».

Une unité de synchronisation ou tâche multimédia constitue donc un cas particulier d'application « temps réel mou » dont le profil temporel met en évidence trois dates significatives (la Figure 2.6-d donne un exemple de profil temporel d'une telle tâche). Ces trois dates notées d_1 , n et d_2 sur la Figure 2.6-d spécifient la flexibilité temporelle susceptible d'être supportée par le traitement considéré, et ont respectivement la sémantique suivante:

- une présentation multimédia doit avoir une certaine persistance temporelle d'où la nécessité d'associer à une unité de synchronisation une date minimale de terminaison notée ici d_1 . Cette valeur constitue la date de terminaison au plus tôt du traitement considéré.
- une date de terminaison nominale ou idéale notée ici n est associée à toute unité de synchronisation. On remarquera sur la Figure 2.6-d que pour cette valeur le bénéfice obtenu est maximum. Cependant cette durée

nominale ne peut être généralement garantie pour des raisons d'indisponibilité de ressources ou d'asynchronisme du système. Ainsi la flexibilité temporelle d'une unité de synchronisation multimédia rend ces tâches adaptables aux ressources et dans une certaine mesure compatibles avec des environnements asynchrones.

Média	Discontinuité maximum admissible (ms)
Voix	10
Vidéo	10
Vidéo compressée	1

Tableau 2.3: Les discontinuités admissibles entre les unités de synchronisation de flux continus.

- conformément aux caractéristiques des tâches « temps-réel » une date de terminaison au plus tard ou « deadline » notée ici d_2 est associée à toute unité de synchronisation.

Les limites de la capacité de résolution temporelle humaine permettent de substituer à toute date un intervalle temporel contenant cette date et identifié subjectivement à celle-ci. Ainsi, la notion de continuité temporelle associée aux flux continus peut-elle s'accommoder d'une discontinuité temporelle maximale entre la date de terminaison d'une unité de synchronisation et la date de début de l'unité de synchronisation suivante (Tableau 2.3).

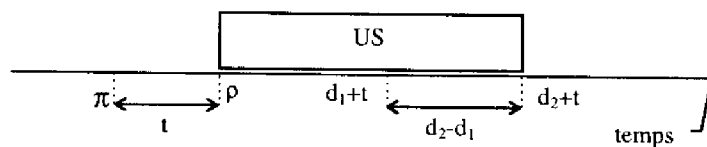


Figure 2.7: Tolérance d'ordonnement et flexibilité temporelle d'une Unité de Synchronisation. La tolérance d'ordonnement de l'unité de synchronisation US à partir de la date au plus tôt d'ordonnement est donnée par t . Les durées d_1 et d_2 définissent la flexibilité temporelle admissible de US .

La permissivité des contraintes « temps-réel » supportée par l'utilisateur ou par une sémantique applicative induit donc une variabilité admissible des dates de début d'activation des unités de synchronisation. Autrement dit à toute unité de synchronisation peuvent être associées une date minimum et une date maximum d'activation. Un ordonnanceur peut donc optimiser le coût d'ordonnement des unités de synchronisation en exploitant à la fois leur flexibilité temporelle et leur flexibilité d'ordonnement (Figure 2.7).

Plus généralement on peut introduire la notion de tolérance de synchronisation d'un événement ou d'une séquence d'événements [Lit94]. Ainsi un événement survenant à une date ρ est synchronisé, à une tolérance temporelle θ près, avec une référence temporelle survenant à la date π , si et seulement si:

$$|\rho - \pi| \leq \theta$$

Cette notion de tolérance de synchronisation s'apparente à la notion de « vecteur de divergence » dans [Rav74].

La notion de robustesse des traitements apparaît également comme une caractéristique fondamentale d'un système « temps-réel ». Dans ce domaine les systèmes multimédias supportent une certaine tolérance aux erreurs de synchronisation; en effet les problèmes de désynchronisation peuvent être compensés dans une certaine mesure en utilisant des techniques de répétition ou de saut d'unités d'information [AH91]. Partant de ces considérations, des traitements d'exceptions réalisant des actions correctives peuvent être associés aux unités de synchronisation afin de préserver la continuité et/ou la sémantique des flux [Ste90].

7. La Synchronisation Intra-flux

La présentation d'un flux exige le respect de contraintes temporelles quant à la date d'activation et la durée des unités de synchronisation réalisant la présentation de ce flux. Nous avons vu que ces contraintes de synchronisation peuvent être naturelles ou synthétiques, mais que dans tous les cas leur respect conditionne la sémantique du flux ou de l'application multimédia. Ainsi la notion de flux est indissociable de celle de schéma de synchronisation. Cette étroite relation entre flux et synchronisation induit que chaque flux d'une application multimédia doit être associé à une spécification des contraintes temporelles devant être satisfaites par l'ensemble des unités de synchronisation réalisant la présentation ou plus généralement le traitement de ce flux. Cette spécification qui doit être donnée par l'auteur de l'application multimédia et mise en oeuvre et contrôlée par l'environnement d'exécution est appelée un *scénario* ou *schéma* de *synchronisation intra-flux*. La synchronisation intra-flux recouvre les contraintes de synchronisation discrètes, continues, naturelles et synthétiques. Conformément à la structure séquentielle des flux, ces schémas de synchronisation mettent en oeuvre uniquement des structures de synchronisation séquentielles. Un schéma de synchronisation intra-flux est donc spécifié par une séquence d'unités de synchronisation $(US_i)_{i \in \{0, \dots, n\}}$ telles qu'à toute unité de synchronisation US_k est associée une suite d'unités d'information $(UI_{k,j})_{j \in \{0, \dots, m^{(k)}\}}$, et un couple (t_k, d_k) définissant la date et la durée d'ordonnement nominales associées à cette unité de synchronisation. La structure de synchronisation séquentielle sous-jacente à la notion de flux impose que : $\forall i \in [0, n-1], t_i + d_i \leq t_{i+1}$. En reconsidérant les contraintes d'ordonnement pour un pipeline de traitements que nous avons mises en évidence dans la Section 6.2 nous en déduisons que toute contrainte de synchronisation intra-flux associée à une étape du pipeline induit des contraintes de synchronisation intra-flux pour les étapes amont du pipeline.

8. La Synchronisation Inter-flux

Une application multimédia met en jeu, par définition, plusieurs flux monomédias. La scénarisation d'une application multimédia exige donc de mettre en oeuvre des schémas de synchronisation faisant intervenir plusieurs flux. On appelle schémas de *synchronisation inter-flux* de tels schémas. A titre d'exemple la Figure 2.8 spécifie à l'aide du paradigme de composition de la ligne temporelle un schéma de synchronisation inter-flux mettant en jeu 3 flux: un flux continu audio, un flux continu vidéo et un flux discret d'images fixes.

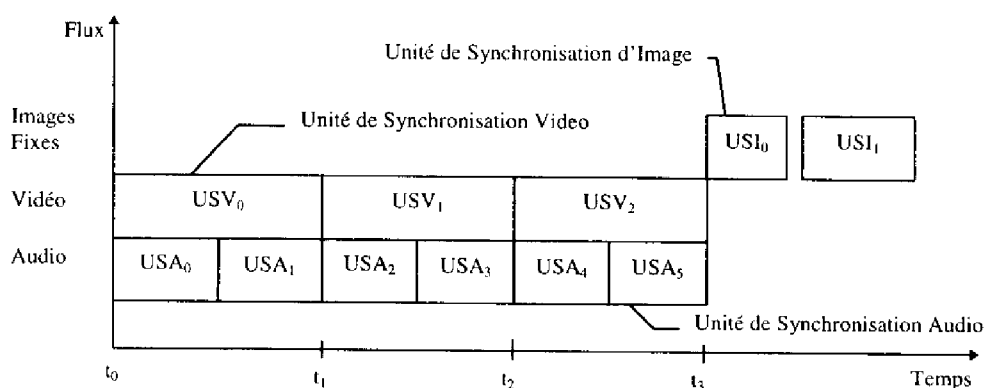


Figure 2.8: Un schéma de Synchronisation Inter-Flux. Ce schéma de synchronisation inter-flux est exprimé à partir du paradigme de composition de la ligne temporelle consistant à définir des scénarios de synchronisation en superposant des flux à un axe temporel.

En terme de synchronisation inter-flux ce scénario spécifie les contraintes suivantes :

1. les traitements des flux audio et vidéo doivent débiter (date t_0) et se terminer (date t_3) en même temps,
2. chaque unité de synchronisation vidéo doit être synchronisée avec les deux unités de synchronisation audio correspondantes,
3. le flux d'images fixes doit être traité dès que le traitement des flux audio et vidéo est terminé.

La synchronisation inter-flux est donc mise en oeuvre à partir de structures de synchronisation séquentielles comme dans le cas de la troisième contrainte de synchronisation, ou bien, à la différence de la synchronisation intra-flux, à partir de structures de synchronisation parallèles comme dans le cas de la première et de la deuxième contrainte. De plus, la deuxième contrainte illustre le concept de *synchronisation continue*. En effet on appelle synchronisation continue tout schéma de synchronisation inter-flux mettant en jeu des points de synchronisation appliqués périodiquement entre les unités de synchronisation de flux continus. La mise en oeuvre de tels schémas de synchronisation s'avère nécessaire de façon à éviter des dérives temporelles entre des flux devant être synchronisés. En particulier, un flux vidéo présentant un locuteur doit être continuellement synchronisé avec le flux audio correspondant. L'intérêt de ce cas particulier de synchronisation continue est bien reconnu par les spectateurs de films ayant fait les frais d'une mauvaise post synchronisation sonore et est désigné dans la communauté de l'industrie vidéo par la dénomination « synchronisation de lèvres » (« lip-synchronization »).

La période entre les points de synchronisation inter-flux spécifiés dans un schéma de synchronisation continu induit la notion de *granularité de synchronisation inter-flux*. La granularité des schémas de synchronisation inter et intra-flux apparaît comme un paramètre induisant une certaine adaptabilité des objets multimédias aux ressources disponibles. Ainsi tout schéma de synchronisation résulte d'un compromis entre les ressources consommées et la qualité de service offerte à l'utilisateur.

9. La Synchronisation Inter-Sites

Si on considère qu'un scénario de synchronisation peut mettre en jeu plusieurs sites clients et serveurs d'unités d'information, c'est à dire si l'on se place dans le cadre général des systèmes multimédia distribués, alors, aux contraintes de synchronisation inter et intra-flux s'ajoutent des contraintes de synchronisation devant être satisfaites par l'ensemble des sites impliqués dans un scénario de synchronisation.

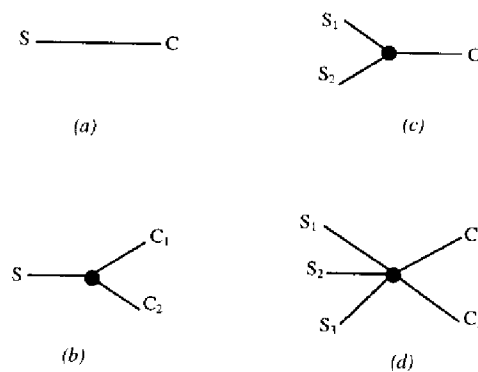


Figure 2.9: Les schémas de synchronisation inter-sites. En (a) un schéma de synchronisation de type 1-1 entre un serveur (*S*) et un client (*C*). En (b) un schéma de synchronisation de type 1-*n*. En (c) un schéma de synchronisation de type *n*-1. En (d) un schéma de synchronisation de type *n*-*m*.

On peut distinguer les schémas de synchronisation inter-sites suivants (du plus élémentaire au plus complexe):

- les schémas de synchronisation 1-1 (Figure 2.9-a). Ces schémas de synchronisation consistent en la production d'un flux sur un site et en sa consommation sur un autre site. Ces schémas de synchronisation sont donc constitutifs de tout schéma de communication/synchronisation de type client-serveur. Des mécanismes de compensation des délais de transfert et du débit disponible, reposant sur la bufferisation et le préchargement des unités d'information permettent de garantir le respect des contraintes de synchronisation intra-flux associées aux schémas de synchronisation 1-1 [LG90c]. Il est à noter cependant que de tels mécanismes induisent un temps de latence au niveau de la restitution qui peut s'avérer incompatible avec une utilisation interactive.

- les schémas de synchronisation 1-n (Figure 2.9-b). De tels schémas correspondent au traitement synchronisé sur plusieurs sites clients d'un flux issu d'un site serveur. Ces schémas rendent nécessaire la mise en oeuvre d'une diffusion partielle à sémantique synchrone ou faiblement synchrone [Tan92].
- les schémas de synchronisation n-1 (Figure 2.9-c). Ces schémas de synchronisation consistent en la synchronisation sur un site client du traitement de plusieurs flux issus de divers sites serveurs. Dans ce cas, aux contraintes de synchronisation intra-flux évoquées pour la synchronisation de type 1-1 s'ajoutent des contraintes de synchronisation inter-flux. Les dérives temporelles inter-flux peuvent être également compensées par un mécanisme de bufferisation sur le site client [Lit93].
- les schémas de synchronisation n-m (Figure 2.9-d). Ces schémas de synchronisation consistent en la synchronisation sur plusieurs sites clients de plusieurs flux issus de divers sites serveurs; à ce titre les 3 types de synchronisation précédents constituent un cas particulier de synchronisation n-m.

Dans tout schéma de synchronisation inter-sites il est nécessaire d'éviter des dérives intempestives entre les horloges des divers sites clients et serveurs impliqués. Les schémas de synchronisation inter-sites sont donc grandement facilités par la mise en oeuvre d'un « temps global » au système distribué multimédia, ainsi que par la disponibilité de mécanismes de diffusion atomique à sémantique faiblement synchrone. La réalisation d'un temps global implique la mise en oeuvre de mécanismes de synchronisation d'horloges des divers sites clients et serveurs impliqués.

10. L'Asynchronisme des Traitements

Lors de la modélisation d'un système il est important, afin de maximiser la généralité du modèle, d'imposer le moins de contraintes possible sur ce système. Dans le cadre général des systèmes multimédias distribués, l'hypothèse d'asynchronisme constitue la contrainte minimale pouvant être utilisée. Cette hypothèse d'asynchronisme postule que les vitesses relatives des processus et des canaux de communication sont non bornées [Sch93]. Une telle hypothèse proscriit la mise en oeuvre d'horloges synchronisées et d'un temps global au système [BM93]. Ainsi cette hypothèse d'asynchronisme est incompatible avec les contraintes temporelles devant être satisfaites par les unités de synchronisation d'une application multimédia distribuée.

Inversement l'utilisation de l'hypothèse de synchronisme fort, située à l'autre extrémité de l'échelle des contraintes, offre un cadre compatible avec la nécessité de mettre en oeuvre des schémas de synchronisation temporellement contraints, mais constitue cependant un hypothèse irréaliste. En effet, si nous considérons le pipeline de traitements distribués illustrés par la Figure 2.3, nous constatons que chaque étape du traitement constitue une source potentielle d'asynchronisme. Or il est pratiquement impossible de mettre en oeuvre à une large échelle un système distribué purement synchrone [Tan92][SH94]. De plus la mise en oeuvre d'un système respectant l'hypothèse de synchronisme impose d'avoir une connaissance a priori et totale sur le comportement du système et celui de l'environnement. Même à coût élevé, la probabilité d'avoir une telle connaissance est très faible [Jen94]. Ainsi, ces considérations nous conduisent à considérer un système distribué multimédia comme un système *faiblement synchrone*, c'est à dire comme un système qui garantit que tous les événements durent un temps fini et qu'ils apparaissent dans le même ordre pour tous les sites participants [Tan92]. Nous allons justifier

et expliciter cette hypothèse de synchronisme faible en étudiant de façon plus détaillée les principales sources potentielles d'asynchronisme dans le pipeline des traitements susceptible d'être parcouru par un flux multimédia, à savoir:

- le système de communication
- les horloges
- le matériel
- le système d'exploitation

10.1 L'Asynchronisme du Système de Communication

Les débits offerts par les réseaux se sont conjugués aux progrès réalisés en matière de compression des données et aux performances des nouvelles générations de microprocesseurs et contrôleurs d'Entrées-Sorties pour permettre d'étendre les paradigmes traditionnels de communication inter-tâches aux données multimédias. C'est ainsi qu'il devient envisageable de développer des applications multimédias distribuées en utilisant le modèle client serveur, la communication par message ou les RPC. Cependant on doit noter que ces paradigmes font totalement abstraction des contraintes de synchronisation intrinsèques aux flux multimédias, et en ce sens sont inadaptés au développement d'applications multimédias distribuées et interactives [CBDW92]. De plus la définition de nouveaux paradigmes de communication intégrant les contraintes de synchronisation des flux multimédias se heurte aux limitations des protocoles de transport actuellement utilisés tels que TCP, UDP ou le Transport ISO [Dan92]. Ainsi, en tout premier lieu, de nouveaux protocoles de transport prenant en charge les contraintes de synchronisation intra-flux, inter-flux, et inter-sites doivent être développés.

La problématique de la synchronisation inter-sites peut être résolue par l'intermédiaire de primitives de diffusion. Des primitives de diffusion ayant une sémantique purement synchrone s'avèrent impossibles à construire effectivement; aussi seules des formes dégénérées offrant des sémantiques faiblement synchrones ou virtuellement synchrones telles que la primitive de diffusion atomique proposée dans le système ISIS [BJ87] sont disponibles pour réaliser des schémas de synchronisation inter-sites. Par définition, les sémantiques de diffusion faiblement synchrones ou virtuellement synchrones induisent une variabilité temporelle dans les schémas de synchronisation inter-sites.

La synchronisation inter-flux est traditionnellement traitée selon des méthodes pouvant être classées en trois groupes [SS93]:

- le multiplexage de flux sur une même connexion [NS92], [HS91],
- les canaux de synchronisation [CCGH92], [SS90]
- les estampilles temporelles [EDP92], [LG91b]

La cohérence temporelle des schémas de synchronisation inter-flux et inter-sites repose cependant en partie sur la disponibilité de mécanismes permettant de satisfaire les contraintes de synchronisation intra-flux. Le respect

service probabilistes, basés sur une caractérisation statistique du trafic, introduisent une variabilité (compatible avec les contraintes de synchronisation des applications « temps-réel multimédia ») dans la qualité de service offerte par un service transport mais permettent en échange un meilleur taux d'utilisation du service de communication [AKRS94]. Ainsi des considérations pragmatiques militent en faveur d'approches probabilistes ou « best-effort » pour la gestion de la qualité de service offerte aux applications multimédias par le niveau transport. Ces considérations sont renforcées par le caractère asynchrone des couches sous-jacentes aux couches transport (réseau, liaison de données) des réseaux locaux ou grandes distances actuellement utilisés. Ainsi les réseaux locaux de type Ethernet, « Token-ring » ou FDDI exhibent des comportements asynchrones ou faiblement synchrones.

De même, l'ATM annoncé comme le futur mode de communication unificateur pour réseaux locaux et grande distance est intrinsèquement une méthode d'accès asynchrone. Ainsi le multiplexage statistique des cellules mis en jeu dans les commutateurs ATM est susceptible d'induire la perte de cellules ainsi que la variabilité des délais et des gigue de transmission des cellules. Les couches AAL (ATM Adaptation Layer) de type 1 et 2, prévues pour offrir une qualité de service compatible avec les flux continus à débit constant et à débit variable respectivement, ne permettent pas cependant de résoudre totalement la problématique de la synchronisation intra-flux au niveau du système de communication, en effet:

- il semble utopique de penser que l'avènement du « tout ATM » est pour un futur proche. En effet, il est fort probable que les autres types de réseaux coexisteront encore pendant de longues années avec l'ATM. Les systèmes distribués multimédias de demain seront donc constitués par des réseaux hétérogènes; ainsi l'utilisation d'un réseau ATM en tant que maillon d'un système de communication mêlant d'autres types de réseaux ne permet pas à elle seule de résoudre la problématique de la synchronisation intra-flux.
- il semble irréaliste de mobiliser des ressources basées exclusivement sur le débit crête d'une connexion ATM de façon à avoir un comportement purement déterministe d'une connexion ATM. Les solutions actuellement prônées tendent à privilégier le débit moyen et la sporadicité d'une connexion pour le contrôle d'accès au réseau. Ainsi les couches d'adaptation ne préservent pas totalement les applications multimédias de pertes ou retards intempestifs de cellules.
- les mécanismes de synchronisation d'horloges entre émetteurs et récepteurs sont tous confrontés à un problème de synchronisation d'horloges distribuées qui induit comme nous le verrons dans la section consacrée à cette problématique une variabilité des schémas de synchronisation inter et intra-flux.

En résumé il apparaît que, pour des raisons essentiellement économiques, les utilisateurs privilégieront l'utilisation de services « faiblement synchrones » dans la nouvelle génération de protocoles et de réseaux adaptés à la transmission de flux multimédias. Autrement dit, ces futurs systèmes de communication pourront exploiter la flexibilité temporelle des flux multimédias à travers des profils de qualité de service de types statistiques ou « best-effort ». Ainsi l'asynchronisme ne sera pas résolument banni des futurs systèmes de communication mais il sera l'objet d'une négociation entre l'utilisateur et le service de transport.

des contraintes de synchronisation intra-flux dans des réseaux de type commutation de paquets ou de cellules passe par un mécanisme de réservations de ressources de communication. En effet, étant donné le caractère temps réel des flux multimédias, en préalable de tout ordonnancement du traitement d'un flux une phase de contrôle d'admissibilité doit intervenir de façon à vérifier que l'ensemble des ressources nécessaires à ce traitement sont disponibles [Buf94], [DTH92]. Ce contrôle d'admissibilité nécessaire en traitement centralisé est rendu plus complexe lorsque les gestionnaires de ressources sont multiples et répartis. Cette répartition rend nécessaire la mise en oeuvre de protocoles de réservation de ressources permettant de réaliser un contrat entre l'application et le service de transport [Fer92,94], [ATWG90]. Les termes de l'engagement contractuel entre le client et le fournisseur du service reposent sur la notion de Qualité de Service (QoS). L'utilisation de la notion de QoS pour spécifier de façon unifiée les ressources nécessaires à tous les niveaux d'un système multimédia offre un paradigme général pour la gestion des ressources [Buf94]. Ainsi, le client d'un service transport décrit ses exigences en matière de qualité de service à travers des paramètres tels que [Fer94]:

- le débit minimum admissible
- le délai maximum admissible
- la gigue maximum admissible
- le taux de perte de paquets maximum suite à des débordements de buffers ou des dépassements de délais de communication

D'un point de vue du service transport ou réseau, la classe des applications multimédias est une sous-classe de celle des applications « temps-réel ». Conformément à l'approche Tenet [Fer94], le problème de la communication de données multimédias doit donc être élargi à celui de la communication dans un contexte « temps réel multimédia » au sens où nous l'avons défini précédemment. Dans le domaine de la communication « temps-réel multimédia » la donnée de paramètres de qualité de service déterministes paraît inadaptée, en effet:

- nous avons vu que les données multimédias supportent une certaine flexibilité en matière de contraintes de synchronisation, il paraît donc souhaitable d'un point de vue économique d'exploiter cette caractéristique au niveau du système de communication;
- l'ensemble des conditions permettant d'assurer la prédictibilité d'un mécanisme de synchronisation est toujours une abstraction et une réduction de la réalité.

Ainsi il est souhaitable de spécifier les paramètres de Qualité de Service (QoS) en combinant une approche déterministe et une approche « best-effort » [Dant92]. Une telle approche permet de plus une gestion plus souple et optimale des ressources du système de communication. Cette approche hybride est réalisée en associant un intervalle de validité à chaque paramètre de qualité de service. La valeur minimum de l'intervalle spécifie la valeur en deçà de laquelle le service obtenu serait inutile, ainsi le respect d'au moins cette valeur est-il nécessaire. La valeur maximum spécifie quant à elle la valeur « best-effort » au delà de laquelle l'utilisateur n'obtient aucun bénéfice ou n'est plus décidé à payer. Ainsi, en précisant des intervalles de validité pour les paramètres de qualité de service d'une connexion transport, le service transport peut adapter les ressources mobilisées pour la connexion en fonction de la charge du réseau [Fer94]. De même, des paramètres de qualité de

10.2 L'Asynchronisme des Horloges

La mise en oeuvre de schémas de synchronisation inter-sites ainsi que les mécanismes de synchronisation inter et intra-flux à base d'estampilles temporelles exigent de disposer d'une horloge globale à l'ensemble des systèmes mis en jeu. Cependant, une horloge, notée C , donnant une valeur $C(t)$ à l'instant UTC t , est exposée à une dérive maximum intrinsèque ρ impliquant la relation suivante:

$$\forall t_1, t_2 \quad (1 - \rho)(t_2 - t_1) \leq C(t_2) - C(t_1) \leq (1 + \rho)(t_2 - t_1)$$

Cette dérive est, pour les horloges à quartz communément utilisées, de l'ordre de 10^{-5} à 10^{-6} secondes. Ainsi au bout d'un temps Δt deux horloges possédant la même dérive maximum peuvent être désynchronisées au plus de $2\rho\Delta t$ unités de temps. Deux horloges C_1 et C_2 vérifient donc la relation suivante à l'instant t :

$$-2\rho t \leq C_1(t) - C_2(t) \leq 2\rho t$$

Ainsi, si l'on veut que ces deux horloges soient au plus désynchronisées d'une valeur ϵ il est nécessaire d'avoir une période de resynchronisation t telle que $t \leq \frac{\epsilon}{2\rho}$.

La resynchronisation des horloges peut être réalisée en adoptant une approche logicielle ou matérielle. La solution matérielle, utilisant des boucles d'asservissement de phase des horloges, bien que fournissant une synchronisation précise entre les horloges, ne constitue pas une solution réaliste étant donné le coût induit en matière de connexions et de matériel spécialisé [SR94]. Des solutions matérielles à base de récepteurs radio WWV ou satellites GEOS permettent à moindre coût de maintenir une précision de l'ordre de ± 10 ms par rapport au temps UTC [Tan92]. En utilisant une approche logicielle il n'est pas possible de synchroniser parfaitement des horloges dans un système réparti car cela réclamerait la connaissance exacte des temps de transfert pour chaque message. Ainsi, avec une approche logicielle, la dérive la plus pessimiste pouvant exister entre les horloges est supérieure ou égale à la différence entre les délais de transfert maximum et le délai de transfert minimum. Dans le cas des réseaux synchrones où l'on connaît une borne supérieure sur le temps de transfert des messages il est possible de construire des algorithmes de synchronisation permettant de borner de façon déterministe la dérive entre les horloges de deux sites. Par contre, si l'on ne connaît pas de borne sur les délais de transfert des messages on ne peut garantir que de façon statistique le respect d'une certaine dérive entre les horloges [Ray91]. Dans la pratique les implémentations d'algorithmes de synchronisation d'horloges tels que celui de Cristian [Cri89] ou celui de l'environnement DCE de l'OSF permettent au mieux de maintenir des horloges de sites répartis synchronisées à 5 à 10ms du temps UTC.

Ainsi dans un système distribué toute durée ou toute date est identifiable à un intervalle temporel spécifiant une zone d'incertitude temporelle; à ce titre ces systèmes doivent être considérés, conformément à la définition donnée en début de cette section, comme faiblement synchrones.

10.3 L'Asynchronisme du Système d'Exploitation

Un système multimédia distribué est constitué d'un ensemble de systèmes de communication et d'un ensemble de systèmes d'exploitation coopérant de façon à assurer une qualité de service de bout en bout. Le système

d'exploitation en tant que gestionnaire des ressources physiques et logiques offertes aux applications multimédias joue donc un rôle fondamental pour la mise en oeuvre et le contrôle de la qualité de service de bout en bout. Ainsi la notion de qualité de service utilisée dans le cadre des protocoles de communication doit être prolongée et intégrée dans les systèmes d'exploitation. Cependant les systèmes d'exploitation actuellement utilisés tels que Unix ou Mach ne sont pas en mesure de gérer efficacement les contraintes temporelles, les hauts débits et le fractionnement des données induits par les flux multimédias [Gov92]. En effet on peut observer les limitations suivantes au niveau des systèmes d'exploitation courants:

- ces systèmes d'exploitation ont une forte orientation temps partagé visant à travers des ordonnanceurs de type « multi-level feedback queue », à privilégier les processus interactifs ainsi que le débit moyen du système d'exploitation [Bac86].
- le caractère non préemptif ou localement préemptif des noyaux induit un temps de latence non déterministe pour le traitement des interruptions. De plus les extensions « temps réel » à UNIX ne permettent généralement pas un contrôle dynamique de l'ordonnancement [BvL91].
- la notion de processus lourd telle qu'elle est utilisée dans Unix induit une surcharge en matière d'ordonnancement incompatible avec le haut degré de parallélisme, les hauts débits et la fractionnement des données associés aux flux multimédias.
- les mécanismes de communication traditionnels induisent des copies entre espaces d'adresses trop coûteux en temps.

Ainsi les systèmes d'exploitation actuels ne mettent en oeuvre aucun mécanisme de réservation de ressources permettant de garantir une qualité de service quelconque à une application multimédia, ils utilisent une stratégie « least-effort » induisant un comportement asynchrone des unités de synchronisation.

Les travaux réalisés dans le cadre de RT-Mach [Tok94], de l'architecture ANSA [CBDW92] [CBR94], ou du projet DASH [ATWG90] préfigurent les caractéristiques des futurs systèmes d'exploitation distribués multimédias. Parmi les solutions et mécanismes envisagés pour la conception de ces systèmes on peut retenir les choix consensuels suivants:

- ces futurs systèmes seront basés sur une architecture micro-noyau. Les micro-noyaux jouant le rôle de véritables « bus logiciels » permettent de reporter dans des processus serveurs des services de communication, de gestion de fichiers, voire des systèmes d'exploitation en entier. Les architectures micro-noyaux renforcent la structuration, l'évolutivité et la robustesse des systèmes d'exploitation. Cependant les micro-noyaux en utilisant systématiquement une communication par messages entre clients et serveurs introduisent généralement une dégradation de performances par rapport à une architecture système monolithique. Ainsi bien qu'apportant des avantages indiscutables les architectures micro-noyaux à elles seules ne sont pas en mesure de résoudre les problèmes soulevés par la gestion des flux multimédias. En particulier des mécanismes de communication inter-tâches performants permettant de partager des structures de données entre espace utilisateur et espace noyau du type de celui proposé dans [Gov92] doivent être utilisés.

- la politique de gestion des ressources pratiquée par le système d'exploitation doit être basée sur la notion de qualité de service. Pour que le système d'exploitation s'adapte à l'évolution de la charge, on doit utiliser des mécanismes permettant une gestion dynamique de la qualité de service. Cette gestion dynamique de qualité de service repose sur des intervalles d'évolution admissible des paramètres de qualité de service spécifiés par l'application [CBDW92] [Tok94].
- le concept de thread doit être étendu à celui de thread temps-réel de façon à permettre un ordonnancement des threads prenant en compte leurs contraintes « temps-réel ». Etant données les capacités de traitement de flux continus offertes par les micro-processeurs 64 bits de la nouvelle génération [Hal94a,94b] [Way94] [TR94] et les avantages associés, en terme de portabilité et d'évolutivité à une solution « tout-logiciel », la notion de thread temps réel est le support de programmation de base pour les systèmes d'exploitation multimédias. A ce titre l'ordonnancement de threads « temps réel » constitue une activité fondamentale et critique pour les systèmes d'exploitation multimédias. L'ordonnancement des threads est d'autant plus critique qu'il conditionne les performances des implantations de protocoles « multi-threadés » basées sur les streams Unix System V [DJ92]. Les caractéristiques dynamiques des applications multimédias rendent nécessaire l'utilisation d'ordonnanceurs dynamiques préemptifs. Cependant les algorithmes d'ordonnancement dynamiques préemptifs traditionnels tels que Rate Monotonic ou Earliest Deadline First imposent des contraintes très limitatives sur les caractéristiques des tâches devant être ordonnancées [Kop93]. Ainsi, il n'existe pas encore de résultats pratiques concluants mettant en évidence des algorithmes d'ordonnancement dynamiques, prédictibles et flexibles qui prennent en compte les contraintes du monde réel [RS94]. De façon à permettre un ordonnancement plus souple des unités de synchronisation, leurs caractéristiques « temps réel multimédia » doivent être exploitées à travers des paramètres de qualité de service de types « best-effort » ou statistiques [LMM92]. Ainsi, la notion de test d'ordonnancabilité doit être remplacée par la notion de test de « présentabilité » basée sur les domaines de variation admissible des paramètres de qualité de service de type « best-effort » [Tok94]. En particulier, la flexibilité et la variabilité temporelle des unités de synchronisation multimédias permettent un ordonnancement plus souple des unités de synchronisation, et par là même devraient faciliter le développement d'algorithmes d'ordonnancement plus généraux calculables avec une complexité « traitable ».
- Les systèmes d'exploitation multimédias doivent permettre une gestion temporisée des entrées-sorties, des accès mémoires et de la communication. Ainsi la communication, les entrées-sorties et les accès mémoires doivent être intégrés dans la politique d'ordonnancement [Tok94], [CBR94]. Encore une fois la variabilité temporelle des unités de synchronisation multimédias peut être exploitée afin de minimiser le coût d'ordonnancement des opérations d'Entrée-Sortie et de communication.

En résumé, les futurs systèmes d'exploitation multimédias ne banniront pas l'asynchronisme mais le géreront à travers des ordonnancements de type « best-effort » exploitant les profils temporels admissibles des unités de synchronisation multimédias.

10.4 L'Asynchronisme du Matériel

Les limitations intrinsèques au matériel fixent les limites inférieures en matière de synchronisme que l'on est en droit d'espérer pour un système multimédia, en l'absence de mécanismes de compensation. Les nouvelles architectures de processeurs et d'ordinateurs universels mettent en oeuvre des architectures asynchrones. En effet la nouvelle génération de micro-processeurs intègre des caches d'instructions et de données multi-niveaux qui génèrent des temps d'accès mémoire et de commutation de contexte non déterministes [SR94]. De même la gestion de mémoire virtuelle introduit une variabilité des temps d'accès mémoire. Cette variabilité est renforcée dans les architectures multiprocesseurs par les mécanismes de réactualisation des caches et les contentions pour l'accès aux bus. De plus les systèmes de gestion d'interruptions hiérarchisées induisent des temps de latence non déterministes. Les disques, les processeurs spécialisés de compression/décompression et les périphériques multimédias sont également autant de sources d'asynchronisme [RW94][Red94]. Ainsi les futures architectures systèmes universelles qui seront les structures d'accueil privilégiées pour les applications multimédias de demain ne seront pas, de par leur caractère « ouvert », des architectures purement synchrones.

11. La Prise en Compte de l'Asynchronisme

11.1 La Notion d'Intervalle de Validité Temporelle

Nous avons vu dans la section précédente que chaque unité de synchronisation est exposée au comportement faiblement synchrone du pipeline des traitements multimédias. De plus, il suffit que l'une au moins des étapes d'un « pipeline » de traitements soit source d'asynchronisme pour que le système ait potentiellement un comportement asynchrone. Il convient également de noter que les variations temporelles et les erreurs sont cumulées lors de la traversée du pipeline. Ainsi dans ce contexte il est totalement irréaliste de vouloir spécifier simplement à l'aide d'une durée nominale la sémantique temporelle d'une unité de synchronisation. En effet, dans un système distribué asynchrone les durées de traitement sont non bornées; cependant une application ne peut se satisfaire des durées de traitement non bornées induites par un système purement asynchrone. D'où l'intérêt de la notion de système faiblement synchrone, qui peut être considéré comme un système asynchrone sur lequel on induit, par l'intermédiaire d'une fonction de contrôle, une variabilité temporelle admissible à toute unité de synchronisation. Cette variabilité temporelle admissible tient compte à la fois de la flexibilité temporelle intrinsèque aux unités de synchronisation multimédias et de l'indéterminisme temporel inhérent aux systèmes asynchrones. Dans un système faiblement synchrone la durée nominale d'un traitement, ou plus généralement d'un délai, doit donc être remplacée par un intervalle temporel spécifiant la variabilité temporelle admissible de l'unité de synchronisation considérée. Cet intervalle, appelé *intervalle de validité temporelle*, spécifie la plage admissible des valeurs pouvant être prises par la durée d'exécution d'une unité de synchronisation sans remettre en cause sa sémantique applicative. Bien évidemment la durée nominale d'exécution appartient à l'intervalle de validité temporelle. Ainsi, l'intervalle de validité temporelle d'une unité de synchronisation peut être spécifié par un triplet (d_1, n, d_2) tel que:

- d_1 représente la durée minimum admissible de l'unité de synchronisation,
- n est la durée nominale de l'unité de synchronisation, c'est à dire la durée idéale souhaitée,

- d_2 représente la durée maximum admissible de l'unité de synchronisation.

Il est important de noter que la notion d'intervalle de validité temporelle permet de décrire de façon unifiée les contraintes temporelles d'une unité de synchronisation, pour tout le continuum des systèmes partant des systèmes purement synchrones et allant jusqu'aux systèmes purement asynchrones,. En effet les contraintes temporelles d'une unité de synchronisation peuvent être décrites:

- dans le cas d'un système synchrone, par un intervalle de validité temporelle réduit à une point, i.e. la durée nominale, (n, n, n) ,
- dans le cas d'un système faiblement synchrone, par un intervalle borné, (d_1, n, d_2) ,
- dans le cas d'un système asynchrone, par un intervalle non borné, i.e. $(d_1, n, +\infty)$,

De plus un intervalle de validité temporelle peut être associé à une fonction de coût permettant ainsi de déterminer le gain obtenu suite à une opération d'extension ou contraction réalisée par un formateur temporel exploitant la flexibilité temporelle d'une unité de synchronisation [BZ93b].

11.2 La Notion de Gigue

On appelle *gigue* la variation de durée nominale d'une unité de synchronisation [LG90a]. Cette gigue est non bornée dans un système asynchrone; c'est pourquoi il convient dans ces systèmes d'introduire la notion de gigue admissible c'est à dire de gigue pouvant être supportée par l'application sans remettre en cause la sémantique applicative.

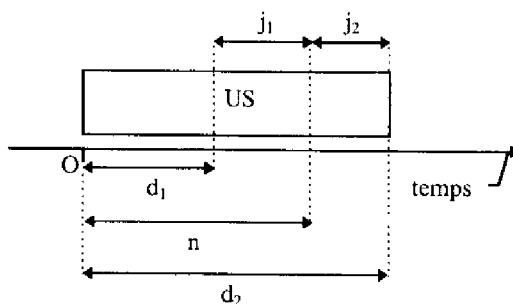


Figure 2.10: La Gigue d'une Unité de Synchronisation. La gigue d'une Unité de Synchronisation *US* est donnée par le couple (j_1, j_2) tel que $j_1 = d_1 - n$ et $j_2 = d_2 - n$, avec (d_1, n, d_2) l'intervalle de validité temporelle associé à *US*.

Ainsi la gigue admissible d'une unité de synchronisation (Figure 2.10) peut être spécifiée à partir d'un couple (j_1, j_2) tel que:

- j_1 appelée *gigue négative admissible* représente la variation minimale admissible de la durée nominale. Ainsi si d_1 et n représentent respectivement les durées minimale et nominale d'une unité de synchronisation la gigue négative admissible de cette unité de synchronisation est donnée par $j_1 = d_1 - n$

- j_2 appelée gigue positive admissible représente la variation maximale admissible de la durée nominale. Ainsi si d_2 et n représentent respectivement les durées maximale et nominale d'une unité de synchronisation la gigue positive admissible de cette unité de synchronisation est donnée par $j_2=d_2-n$

Les schémas de synchronisation intra-flux et inter-flux précédemment évoqués doivent donc intégrer la variabilité temporelle admissible des unités de synchronisation; à ce titre les durées nominales des unités de synchronisation doivent être remplacées par des intervalles de validité temporelle. Cependant, la gigue réelle subie par les unités de synchronisation d'un flux peut, étant donné le caractère asynchrone du système sous-jacent et malgré la variabilité temporelle des unités de synchronisation introduite par les intervalles de validité temporelle, remettre en cause les schémas de synchronisation intra-flux. Le rôle de la fonction de contrôle de la synchronisation, caractérisant les systèmes faiblement synchrones, est alors de forcer le respect des schémas de synchronisation et d'appliquer éventuellement des actions correctives. Ainsi, par exemple, lorsque les contraintes de synchronisation intra-flux associées à des flux continus ne peuvent être satisfaites à cause de retards intempestifs, dus à la traversée du pipeline de traitements, on peut exploiter la « localité temporelle » de certains médias, en particulier la vidéo, de façon à masquer les discontinuités du flux. Dans ce cas, la duplication d'une unité de synchronisation peut masquer la non disponibilité de la suivante tout en accroissant cependant la latence du flux au risque de ne plus être compatible avec les délais maximum admissibles dans le cadre d'une utilisation interactive. Inversement la latence d'un flux peut être réduite en « sautant » des unités de synchronisation [JSS94]. Cependant, si les flux vidéo se prêtent pas à des mécanismes de « saut » et « duplication », il n'en est pas de même des flux audio ou textuels, de plus la sémantique applicative peut être incompatible avec tout mécanisme de synchronisation susceptible d'entraîner une perte d'information.

11.3 La Notion de Dérive

Les giges des unités de synchronisation composant un flux ont un effet cumulatif. On appelle dérive d'un flux la somme, sur une période donnée, des giges des unités de synchronisation composant ce flux (Figure 2.11).

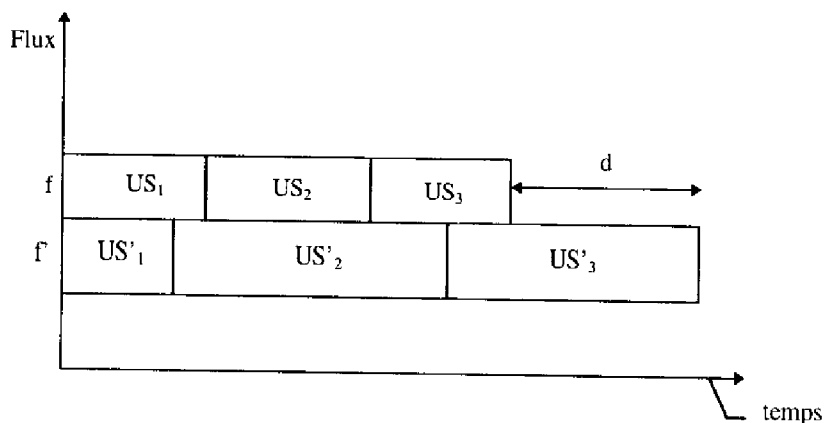


Figure 2.11: La dérive d'un flux. f représente un flux dont les unités de synchronisation n'ont subi aucune gigue (i.e. ont une durée égale à leur durée nominale). f' représente le même flux dans le cas où les unités d'information le composant subissent une gigue. La durée d représente alors la dérive entre le flux parfait f et le flux f' perturbé par la gigue.

Ainsi, la dérive des flux composant un scénario de synchronisation peut remettre en cause la cohérence des schémas de synchronisation inter-flux. Autrement dit le non déterminisme temporel des unités de synchronisation induit un comportement non-déterministe des schémas de synchronisation inter-flux. De façon à supprimer cet indéterminisme de synchronisation il est donc nécessaire d'associer à la problématique de la synchronisation inter-flux une sémantique de synchronisation tenant compte de la dérive des flux. Cette sémantique de synchronisation peut alors servir de base à la mise en oeuvre d'une fonction de contrôle de la synchronisation inter-flux dans un système multimédia [CBDW92].

Des études expérimentales, en particulier celles exposées dans [SE93], ont mis en évidence les niveaux de tolérance admissibles par un utilisateur en matière de dérive entre les divers flux impliqués dans un schéma de synchronisation inter-flux. Ces niveaux de tolérance sont résumés par le tableau 2.4.

Media		Mode	Dérive Admissible
vidéo	animation	corrélé	+/-120ms
	audio	synchronisation des lèvres	+/-80ms
	image	overlay	+/-240ms
		non overlay	+/-500ms
	texte	overlay	+/-240ms
		non overlay	+/-500ms
audio	animation	événements corrélés	+/-80ms
	audio	étroitement couplé (stéréo)	+/-11µs
		faiblement couplé (dialogue avec plusieurs participants)	+/-120ms
		très faiblement couplé (musique d'ambiance)	+/-500ms
	image	étroitement couplé (musique et notes associées)	+/-5ms
		faiblement couplé (diapositives et bande son associée)	+/-500ms
	texte	annotation textuelles	+/-240ms
	pointeur	audio associée à un objet désigné	-500ms, +750ms

Tableau 2.2: Les dérives inter-flux admissibles (extrait de [SE93])

Les contraintes de synchronisation inter-flux peuvent donc s'accommoder, dans une certaine mesure, de la flexibilité temporelle des unités de synchronisation multimédias. Ainsi, étant donné que la dérive inter-flux permet une gestion plus économique des ressources et conditionne la sémantique applicative, cette notion apparaît comme une composante fondamentale de tout scénario de synchronisation multimédia. Autrement dit, toute sémantique de synchronisation multimédia doit, pour être complète, intégrer de façon précise à la fois les concepts de gigue et de dérive. On notera que les mécanismes de « saut » et « duplication » susceptibles d'être exploités afin de résoudre des problèmes de désynchronisation intra-flux peuvent remettre en cause des schémas de synchronisation inter-flux. Les traitements d'exceptions associés à des problèmes de désynchronisation inter-flux peuvent mettre en oeuvre des mécanismes de synchronisation adaptatifs jouant, sans entraîner de perte d'information, sur la vitesse de restitution relative des médias [FSMNM93]. Cependant de tels mécanismes

remettent en question les contraintes de synchronisation intra-flux. Inversement, les mécanismes de synchronisation adaptatifs proposés dans [Lit93] s'attachent à réguler les avances/retards entre flux à l'aide de « sauts » et « duplications » réalisés dans les divers flux impliqués.

12. Conclusion

Partant du paradigme de la régie de production et de ses limitations nous avons introduit dans ce chapitre la notion de flux multimédia. Nous avons vu que la notion de flux multimédia recouvre à la fois une structure de données composée d'unités d'informations et une structure de synchronisation composée d'unités de synchronisation. Une taxonomie détaillée des flux nous a permis de mettre en évidence le rôle fondamental joué par le temps dans les flux multimédias. De plus, les contraintes temporelles intrinsèques aux unités de synchronisation multimédias mettent en évidence une nouvelle classe de tâches « temps-réel » pour lesquelles la notion de date de terminaison au plus tard est étendue par celle d'intervalle de validité temporelle. Ainsi les unités de synchronisation multimédias admettent une certaine variabilité temporelle qui s'avère être compatible, dans une certaine mesure, avec l'asynchronisme intrinsèque aux systèmes distribués. La notion d'intervalle de validité est apparue de façon récurrente à travers l'analyse qui a été faite sur les facteurs d'asynchronisme ainsi que sur la flexibilité temporelle des unités de synchronisation. Il a été de plus montré que la notion de scénario de synchronisation recouvrant à la fois les contraintes de synchronisation intra-flux, inter-flux et inter-sites est une notion fondamentale pour la sémantique applicative. Il est ainsi apparu que les notions de gigue et de dérive admissibles doivent être intégrées et explicitées dans un scénario de synchronisation car elles permettent une gestion plus souple et moins coûteuse de l'ensemble des ressources du système distribué multimédia.

Les contraintes de synchronisation d'une application multimédia déterminent en grande partie la qualité de service devant être mise en oeuvre au niveau de l'ensemble des couches du système multimédia. Ainsi, étant donné le caractère fondamental joué par les scénarios de synchronisation et la complexité arbitraire de leur structure, il est nécessaire de disposer d'une méthode permettant de les spécifier rigoureusement. Les cadres semi-formels proposés par le paradigme de composition de flux de la régie de production ou par la notion de ligne temporelle sont apparus à bien des égards trop limités pour permettre une spécification précise et complète de scénarios de synchronisation multimédias dans un cadre asynchrone ou faiblement synchrone.

Partant des considérations que nous venons de résumer, nous nous proposons dans le chapitre suivant de définir un nouveau modèle permettant une spécification précise, complète et aisée de scénarios de synchronisation dans des systèmes multimédias distribués.

Chapitre 3

Modélisation Formelle de la Synchronisation Multimédia

1. Introduction

Le Chapitre 2 a montré que les scénarios de synchronisation mis en oeuvre dans un système multimédia distribué conditionnent en grande partie la sémantique applicative. Il paraît donc important, en accord avec les principes fondamentaux du génie logiciel [Som92], en préalable à la conception d'un système multimédia, de spécifier de façon précise, complète et cohérente les contraintes de synchronisation devant être satisfaites par un système multimédia. Outre sa fonction de support formalisé pour la communication entre spécificateurs et utilisateurs, la spécification des schémas de synchronisation internes à une application multimédia s'avère indispensable pour générer les profils d'ordonnancement (et plus généralement les paramètres de qualité de service) des unités de synchronisation multimédias sous-jacentes à tout traitement multimédia. Des méthodes de spécification de scénarios de synchronisation multimédias sont d'autant plus nécessaires que ces scénarios peuvent être d'une complexité arbitraire et qu'il n'a pas été proposé jusqu'à ce jour de sémantique de synchronisation intégrant le caractère faiblement synchrone des systèmes distribués.

La mise en oeuvre d'une méthode de spécification passe par la définition d'un modèle permettant d'abstraire et de caractériser les informations pertinentes. Le chapitre précédent a mis en évidence que les notions d'unités de synchronisation, de flux, de synchronisation inter et intra-flux, d'intervalle de variabilité temporelle, de gigue et de dérive sont constitutives et fondatrices d'un scénario de synchronisation. A ce titre tout modèle pour la spécification de la synchronisation multimédia doit permettre d'explicitier ces notions.

Le caractère critique des contraintes de synchronisation, la nécessité de clarifier leur nouvelle problématique ainsi que la complexité des nouveaux standards de représentation de documents multimédias tels que MHEG ou Hytime, qui sont avant tout des langages de conception, conduisent à privilégier l'utilisation de techniques de spécification formelle. Parmi les avantages associés à l'utilisation de techniques de spécification formelle on peut noter :

- l'utilisation des techniques de spécification formelle permet une détection au plus tôt des erreurs dans le cycle de vie du logiciel. Ainsi les techniques formelles augmentent la qualité des systèmes tout en diminuant leur coût de développement [Hal90][BH95a,b],
- les techniques de spécification formelle permettent une compréhension et une description plus précises et moins ambiguë d'une problématique. En particulier elles sont susceptibles de fournir une sémantique formelle à la problématique de la synchronisation multimédia. Les scénarios de synchronisation formellement spécifiés peuvent ainsi servir à générer de façon rigoureuse des profils d'ordonnement [CBDW92].
- les techniques de spécification formelle permettent de mettre en oeuvre des méthodes de simulation et d'analyse: la simulation interactive du système peut être utilisée pour acquérir des connaissances complémentaires, alors que la simulation automatique permet de réaliser des mesures de performances et de faire du prototypage; de plus, les méthodes d'analyse permettent de vérifier si le système possède de « bonnes propriétés » logiques et temporelles.

Plus particulièrement les modèles formels peuvent apporter une contribution importante dans le domaine de la réalisation de documents multimédias. En effet, la réalisation de documents multimédias s'inscrit généralement dans un cycle de vie mettant en jeu deux étapes fondamentales:

- Une étape d'édition dans laquelle l'auteur définit les unités d'information du document ainsi que leurs attributs temporels, spatiaux, logiques et sémantiques. De façon à préserver le logiciel multimédia du syndrome du « plat de spaghettis » qui a frappé le logiciel dans ses jeunes années il apparaît nécessaire de réaliser cette étape d'édition selon une approche structurée. En particulier l'utilisation de structures temporelles, qui comme nous l'avons vu sont au coeur de la problématique de la conception d'applications multimédias, semble souhaitable. Deux grandes approches radicalement opposées peuvent être utilisées lors de cette étape. La première approche, qualifiée de concrète, s'attache à expliciter sous une forme procédurale la gestion du flot de contrôle de l'application multimédia. Cette approche transforme l'auteur en programmeur en exigeant de lui d'être au fait de tous les détails d'implantation de l'application multimédia. La deuxième approche, qualifiée d'abstraite, consiste à spécifier le comportement de l'application multimédia à partir d'un modèle formel permettant à l'auteur de s'abstraire des détails d'implantation de l'application et lui imposant un modèle mental pour la réalisation du document. Cette deuxième approche grâce au support formel, abstrait et structural qu'elle fournit à l'auteur est la plus à même de lui permettre de maîtriser la complexité et d'assurer la validité d'une application multimédia.
- Une étape de présentation de l'information à l'utilisateur. Les deux grandes approches susceptibles d'être adoptées lors de la phase d'édition induisent deux façons de mettre en oeuvre cette étape. En effet les documents multimédias édités selon une approche concrète sont apparentés à des programmes qui, suite à une étape de compilation et édition de liens, peuvent être exécutés sur une machine cible. Par contre les documents réalisés selon une approche abstraite doivent faire l'objet d'une étape d'interprétation à partir d'un exécutif parfois appelé formateur¹. Cet exécutif s'apparente à une machine virtuelle assurant l'indépendance

¹ On parle de formateur spatial ou temporel en fonction de la catégorie des structures interprétées par le formateur.

du document par rapport à la machine réelle sous-jacente. La spécification formelle d'un document multimédia peut également faire l'objet d'une traduction automatique vers une structure de document normalisée.

Ces deux étapes sont mises en oeuvre à partir d'un cycle de vie itératif mettant alternativement en jeu édition et présentation. L'adoption d'un modèle formel lors de l'étape d'édition permet d'associer une sémantique formelle au document et par là même de garantir la cohérence du document et de guider l'auteur en supprimant tout risque d'ambiguïté dans les étapes d'édition et de présentation. Un modèle formel offre des capacités d'abstraction permettant à l'auteur de porter en tout premier lieu son effort sur les entités et structures fondamentales du système qu'il modélise. De plus un modèle formel constitue un support privilégié pour mettre en oeuvre des techniques de simulation, de vérification et d'analyse permettant d'établir et de démontrer les propriétés du système ainsi modélisé. L'étape de présentation bénéficie également grandement d'un modèle formel car celui-ci définit précisément le comportement devant être satisfait par le formateur.

L'objet de ce chapitre est donc de définir les grandes lignes d'un modèle formel permettant une spécification complète et précise de la synchronisation dans les systèmes multimédias. Ainsi, après avoir défini les critères de choix d'un modèle, nous montrerons que le formalisme des réseaux de Petri répond en grande partie aux qualités attendues d'un modèle pour la synchronisation multimédia. Cependant, un panorama des extensions temporelles déjà apportées aux réseaux de Petri mettra en évidence des limitations rédhibitoires qui nous conduiront à proposer dans le chapitre suivant un nouveau modèle.

2. Organisation du Chapitre

Ce chapitre identifie dans un premier temps (section 3) les traits fondamentaux qu'il est important de pouvoir modéliser dans un système multimédia. Partant de ces considérations, les caractéristiques générales d'un modèle temporel pour la spécification des contraintes de synchronisation multimédia sont mises en évidence (section 4). La confrontation de ce modèle général aux principales techniques de modélisation susceptibles d'être adoptées pour la modélisation des systèmes multimédias, désigne le formalisme des réseaux de Petri comme le formalisme le plus adapté (section 5). Une étude critique des réseaux de Petri étendus par le temps montre alors les limitations des formalismes proposés jusqu'à ce jour, et souligne la nécessité de proposer une nouvelle extension temporelle comblant les lacunes des modèles préexistants, tout en héritant de leurs acquis théoriques (section 6).

3. Que Modéliser ?

Nous avons vu que la conception d'une application multimédia est grandement bénéficiaire de l'utilisation d'un modèle formel permettant d'adopter une approche structurale et abstraite lors de la phase d'édition, phase qui s'apparente ainsi à une étape de spécification formelle. De façon à assurer la complétude d'un modèle formel et la cohérence de l'étape d'édition, il semble nécessaire de pouvoir, à l'aide d'un même formalisme unificateur, exprimer les structures spatiales, temporelles, logiques et sémantiques susceptibles d'intervenir dans une application multimédia. Les structures spatiales peuvent être considérées comme moins fondamentales car structures temporelles, logiques et sémantiques d'un document multimédia induisent des contraintes sur les

structures spatiales. De plus nous avons vu dans le chapitre précédent le caractère fondamental joué par le temps dans une application multimédia. Cette considération liée au fait que de nombreuses contributions ont été réalisées dans le domaine de la modélisation des structures logiques et sémantiques des documents [BB91][DD94], conduisent à placer les structures temporelles au coeur de la problématique de la modélisation d'applications multimédias. Autrement dit un modèle pour la spécification d'applications multimédias doit être avant tout un modèle temporel. Le Chapitre deux a clairement identifié les caractéristiques devant être offertes par un tel modèle temporel, en particulier il doit permettre:

- de spécifier des scénarios de synchronisation intra ou inter-flux de complexité arbitraire,
- de modéliser l'indéterminisme temporel des schémas de synchronisation inter et intra-flux. Cet indéterminisme temporel résulte du caractère synchrone ou faiblement synchrone des systèmes distribués multimédias. En outre, même dans un cadre synchrone, n'induisant aucune dérive sur les temps de traitements, la flexibilité temporelle des unités de synchronisation multimédias est une caractéristique qu'il est important de spécifier car elle induit une gestion plus souple des ressources du système [Bul93]. Nous avons vu que la laxité temporelle et l'indéterminisme temporel d'une unité de synchronisation multimédia peuvent être spécifiés à partir de la notion d'intervalle de validité temporelle. La notion d'intervalle de validité temporelle s'avère de plus être un précieux auxiliaire méthodologique. En effet, en début de la phase de spécification le spécifieur a une connaissance partielle, imparfaite et imprécise des contraintes temporelles du système qu'il souhaite modéliser. Ainsi le spécifieur utilisera initialement des intervalles de validité temporelle « grossiers » qui seront « affinés » progressivement au cours de la spécification.
- d'exprimer l'indéterminisme de synchronisation susceptible d'intervenir dans des schémas de synchronisation inter-flux. Cet indéterminisme de synchronisation permet d'exprimer la disjonction de plusieurs schémas de synchronisation potentiellement admissibles [WR94]. La notion d'indéterminisme de synchronisation, tout comme l'indéterminisme temporel, est une aide méthodologique permettant au spécifieur d'exprimer la connaissance partielle et imparfaite qu'il peut avoir de certains schémas de synchronisation inter-flux.
- d'exprimer l'adaptabilité d'une application à son environnement. La spécification de comportements adaptatifs s'avère fondamentale pour assurer la portabilité des applications multimédias ainsi que la cohérence de leur fonctionnement dans un environnement faiblement synchrone [CSD95].

De plus la modélisation de structures logiques et temporelles nécessite un modèle ayant respectivement des capacités d'encapsulation, de modularité et d'expression de liens sémantiques entre unités d'information.

4. Comment Modéliser ?

Etant donné le caractère fondamental joué par le temps dans la modélisation multimédia il apparaît qu'un modèle pour la spécification d'applications multimédias doit avant tout proposer un modèle temporel tenant compte des spécificités des unités de synchronisation multimédias. D'une façon générale un modèle temporel s'appuie sur une représentation du temps et met en jeu des unités temporelles de base qui peuvent être soit des instants (*i.e.* des dates) soit des intervalles (*i.e.* des durées) [PL95]. Un modèle temporel doit permettre d'associer des

informations quantitatives et qualitatives (*i.e.* des relations temporelles entre unités temporelles) aux unités temporelles de base. Ces informations temporelles peuvent être exprimées soit à partir d'une représentation absolue du temps soit à partir d'un temps relatif. En associant les unités temporelles à un axe temporel unique, un modèle basé sur un temps absolu induit un ordre temporel total entre les unités temporelles. Ainsi dans un modèle basé sur un temps absolu les relations temporelles entre unités sont implicites. C'est dans cette incapacité à expliciter les relations temporelles entre unités temporelles que résident les limitations des modèles basés sur un temps absolu. En effet ceux-ci n'offrent aucune structure de synchronisation de haut niveau permettant de traiter de façon abstraite une relation de synchronisation entre plusieurs unités de synchronisation. Ainsi est-il difficile de modifier des schémas de synchronisation basés sur un temps absolu tout en préservant la cohérence des schémas de synchronisation inter et intra-flux. Par contre les modèles basés sur un temps relatif présentent des caractéristiques radicalement opposées. En effet ces modèles permettent d'explicitement des relations temporelles qui permettent de déduire de façon implicite un ordre temporel partiel sur les unités temporelles. L'avantage principal de modèles basés sur un temps relatif réside dans leur capacité à structurer et à abstraire des schémas de synchronisation. Ces structures temporelles abstraites sont à la base d'une approche modulaire du processus de spécification. Cependant une approche basée sur un temps relatif n'est pas sans inconvénient. En effet étant donné le caractère itératif du processus de spécification qu'elle sous-tend, elle expose les schémas de synchronisation à des incohérences qualitatives ou quantitatives [All83]. Un modèle basé sur un temps relatif doit donc offrir des techniques de vérification de la cohérence logique et temporelle des schémas de synchronisation qu'il permet de mettre en oeuvre.

De plus les informations temporelles quantitatives de type instants ou intervalles doivent pouvoir être exprimées de façon déterministe ou indéterministe. Nous avons mis en évidence dans le chapitre 2 que la notion d'intervalle de validité temporelle peut être considérée comme une unité temporelle susceptible de décrire aussi bien l'indéterminisme temporel d'un instant que d'un intervalle. La notion d'intervalle de validité temporelle apparaît donc comme l'unité temporelle de base d'un modèle temporel pour la spécification d'applications multimédias. De plus la notion d'intervalle est conforme à la perception abstraite qu'a l'auteur des unités de synchronisation en tant qu'entités perçues dans la durée.

Il apparaît que les informations temporelles quantitatives peuvent être exprimées dans un cadre déterministe à l'aide de trois relations temporelles (*i.e.* avant, après, en même temps) lorsque les unités temporelles de base sont des instants. De même [All83] montre qu'il existe treize façons de combiner temporellement deux intervalles temporels. Etant donné qu'une relation temporelle indéfinie (ou non-déterministe) résulte d'une disjonction de relations temporelles déterministes il existe donc 2^3 relations temporelles indéfinies entre instants et 2^{13} relations temporelles non-déterministes entre intervalles. Considérant la permissivité temporelle des schémas de synchronisation multimédias, [WR94] met en évidence que seulement 4 parmi ces 8 relations temporelles non-déterministes sont suffisantes pour exprimer toute relation temporelle indéterminée entre deux événements: il s'agit des relations « avant », « après », « en même temps », « n'importe quand ». Ces 4 relations temporelles indéterminées sur des événements permettent de construire 29 relations temporelles indéterminées entre des intervalles. Ces 29 relations temporelles sont définies en calculant toutes les conjonctions temporellement cohérentes entre relations temporelles de base mettant en jeu pour chaque intervalle les événements significatifs début d'intervalle et fin d'intervalle. [WR94] en déduit donc que ces 29 relations temporelles constituent

l'ensemble minimum de relations temporelles qui doivent pouvoir être exprimées par un modèle de synchronisation basé sur la notion d'intervalle. En effet, si un modèle ne permettait pas de représenter ces 29 relations temporelles, son pouvoir d'expression serait inférieur à celui du modèle temporel basé sur les 4 opérateurs de synchronisation entre événements.

Ainsi, en résumé, un modèle temporel pour la synchronisation multimédia doit :

- proposer des structures de synchronisation induisant un modèle temporel à base d'un temps relatif,
- utiliser des intervalles comme unités temporelles,
- permettre de modéliser au moins les 29 relations temporelles non-déterministes mises en évidence par [WR94].
- offrir un cadre formel permettant de vérifier les propriétés logiques et temporelles d'un schéma de synchronisation.

5. Choix d'un modèle Formel

De façon à évaluer l'adéquation d'un modèle formel à la spécification d'une problématique il paraît nécessaire de le confronter à trois critères fondamentaux :

- le *pouvoir d'expression* du modèle, ou pouvoir de modéliser de façon complète une problématique donnée. En particulier un modèle pour la synchronisation multimédia doit intégrer un modèle temporel conforme au modèle général se dégageant de la section précédente.
- le *pouvoir d'analyse* du modèle, ou pouvoir de mettre en oeuvre des techniques permettant de vérifier les propriétés d'un système. Les modèles formels apportent dans ce domaine une contribution déterminante en permettant de démontrer les propriétés d'une spécification et en fournissant des fondements rigoureux sur lesquels des opérations de simulation, vérification et analyse peuvent être réalisées.
- le *pouvoir de modélisation* du modèle, ou pouvoir de modéliser facilement une problématique donnée. Ce critère recouvre également le pouvoir d'abstraction du modèle, autrement dit sa capacité à offrir des techniques qui offrent au spécifieur des niveaux d'abstraction, reposant généralement sur des mécanismes d'encapsulation, qui lui permettent de s'affranchir de détails non pertinents et ainsi de maîtriser la complexité d'un système. De plus, le pouvoir de modélisation d'une technique de spécification dépend de son aptitude à proposer au spécifieur des niveaux d'abstraction proches de sa perception du problème. En particulier les formalismes graphiques apportent une contribution importante en matière de pouvoir de modélisation.

Un modèle pour la spécification de scénarios de synchronisation multimédias doit donc offrir « le meilleur compromis » entre ces trois critères.

Le tableau 3.1 synthétise la façon dont les principaux modèles proposés jusqu'à ce jour pour la spécification de systèmes multimédias répondent aux critères fondamentaux qui ont été précédemment introduits. En particulier

ce tableau met en évidence qu'il n'existe pas à ce jour de modèle formel offrant un compromis acceptable entre les trois critères fondamentaux (*i.e.* pouvoirs d'expression, de modélisation et de validation).

Ces modèles pour la plupart sont des modèles « ad hoc » issus de considérations spécifiques aux systèmes multimédias. La forte spécificité multimédia de ces modèles limite leur portée d'utilisation. De plus, la plupart des tentatives de définition de modèles pour la synchronisation multimédia ont ignoré le riche héritage apporté par la communauté scientifique dans le domaine plus général de la modélisation de systèmes temps-réels. Hors, nous avons vu dans le chapitre deux qu'un système multimédia constitue un cas particulier de système temps réel. Plus précisément le chapitre deux a introduit la notion de système faiblement synchrone et a montré que cette notion fournit un cadre conceptuel pour tout type de système temps-réel y compris les systèmes multimédias. Ainsi afin d'assurer un caractère aussi universel que possible à notre contribution, nous nous proposons de définir, en considérant les principaux résultats obtenus dans la modélisation formelle de systèmes temps réels, un modèle formel pour la synchronisation dans les systèmes faiblement synchrones.

	<i>Ligne temporelle</i> [Way91][DD94]	<i>Réseaux temporels de points</i> [BZ93a,b]	<i>Expressions de Chemin</i> [CH74][MHEG]	<i>Vidéo Algébrique</i> [WDG95]
<i>unités temporelles</i>	instants	instants	intervalles	intervalles
<i>modèle temporel</i>	temps absolu	temps relatif	temps relatif	temps relatif
<i>pouvoir d'expression</i>	faible, pas d'indéterminisme temporel ou de synchronisation Absence de structures de synchronisation.	Satisfaisant	faible, pas d'indéterminisme temporel ou de synchronisation Absence de structures de synchronisation.	Non satisfaisant car ne possède pas d'opérateurs permettant d'exprimer l'indéterminisme temporel et de synchronisation
<i>pouvoir d'analyse</i>	bon	limité par l'absence de sémantique formelle	limité par l'absence d'une sémantique formelle.	bon
<i>pouvoir de modélisation</i>	faible, pas de capacité d'abstraction et de modularité	faible, pas de capacité d'abstraction et de modularité	faible, pas de capacité d'abstraction et de modularité	faible, langage algébrique

Tableau 3.1: Evaluation des principaux modèles pour la synchronisation multimédia

Le Chapitre deux a mis en évidence que l'état d'un système multimédia ne change pas continûment dans le temps mais change lors de l'apparition d'événements discrets (*i.e.* le début ou la fin d'une unité de synchronisation, une interruption issue de l'environnement). Ainsi, par définition, un système multimédia appartient à la famille des Systèmes Dynamiques à Événements Discrets (SDED). De nombreux modèles formels ont été proposés pour la modélisation de SDED. Ainsi toute tentative de modélisation formelle de systèmes multimédias doit en tout premier lieu considérer les modèles existants en les confrontant aux critères précités.

Une classification des modèles formels pour les SDED est donnée par le Tableau 3.2 [Yu89]. Etant donné que la modélisation de la synchronisation multimédia nécessite de pouvoir spécifier des contraintes temporelles, seules les méthodes permettant d'explicitier l'évolution du temps sont susceptibles d'offrir un pouvoir d'expression satisfaisant. Si nous examinons la colonne correspondante dans le Tableau 3.2 nous remarquons alors que les Réseaux de Petri (RdP) constituent le seul modèle formel permettant à partir d'un même formalisme d'adopter à la fois une approche logique, algébrique et orientée performance pour l'étude d'un SDED. En effet, de nombreuses extensions temporelles aux réseaux de Petri de base ont été proposées, offrant autant de moyens de spécification des contraintes de synchronisation logiques et temporelles d'un SDED. De plus, les algèbres Min-Max [CMQV89] permettent d'adopter une approche purement algébrique pour l'analyse de réseaux de Petri à structure de graphes d'événements. Enfin, les réseaux de Petri temporisés stochastiques [JY91] permettent d'analyser les performances d'un SDED en associant des lois de distribution probabilistes aux dates d'occurrences d'événements. Il apparaît donc qu'à travers un formalisme unificateur les réseaux de Petri offrent une palette complète de points de vue de modélisation et de méthodes d'analyse ayant de solides bases formelles [Mur89]. L'universalité et l'adaptabilité du formalisme des réseaux de Petri sont confirmées par la grande variété des extensions ayant été apportées au modèle de base, ainsi que par la grande diversité des domaines applicatifs dont ils ont permis la modélisation [Dia82]. Ainsi, grâce à la panoplie complète de techniques d'analyse développées autour des réseaux de Petri, ce formalisme offre un pouvoir d'analyse supérieur à celui d'autres formalismes existants. Il convient cependant de souligner que les techniques d'analyse des réseaux de Petri classiques et étendus reposent principalement sur la génération d'un système de transition étiqueté, on rejoint par là même un formalisme générique sous-jacent couvrant un ensemble de techniques de spécification incluant les réseaux de Petri [Arn92].

	Temporisé	Non Temporisé
Modèle Logique	Logique temporelle Extensions temporelles aux RdP	Automates à Etats Finis Réseaux de Petri
Modèle Algébrique	Algèbre Min-Max RT-Lotos	CCS CSP Lotos
Evaluation des Performances	Chaînes de Markov Réseaux de Files d'Attentes Réseaux de Petri Temporisés Stochastiques	

Tableau 3.2: Classification des modèles formels pour SDED

L'analyse du pouvoir d'expression et de modélisation de l'ensemble des familles de modèles illustrées par le Tableau 3-2 milite également en faveur du formalisme des réseaux de Petri. En effet, il a été montré que les réseaux de Petri étendus (*i.e.* les réseaux de Petri permettant de réaliser le test à zéro) ont le pouvoir d'expression

d'une machine de Turing [Pet81]. Ainsi tout algorithme calculable peut être modélisé sous la forme d'un réseau de Petri étendu. En ce sens on peut dire que les réseaux de Petri étendus ont un pouvoir d'expression maximal.

De plus, le formalisme graphique des réseaux de Petri, souvent perçu de façon réductrice comme leur principal attrait, leur confère un indiscutable avantage potentiel en matière de pouvoir de modélisation, et leurs diverses extensions ont encore accru ce pouvoir de modélisation.

En résumé, les réseaux de Petri apparaissent par leurs pouvoirs d'analyse, de modélisation et d'expression, comme le formalisme le plus abouti pour la modélisation des SDED. Partant de cette considération nous allons dans la section suivante, à travers un panorama des extensions temporelles ayant été apportées aux RdP, étudier l'adéquation des formalismes existants pour la modélisation de la problématique de la synchronisation multimédia.

6. Les Réseaux de Petri et la Synchronisation Multimédia

6.1 Méthodologie de Comparaison

Dans cette section, nous allons confronter à la problématique de la synchronisation multimédia les principales extensions temporelles apportées aux RdP². L'objectif de cette étude est de déterminer l'adaptation de ces modèles à la modélisation de scénarios de synchronisation multimédias.

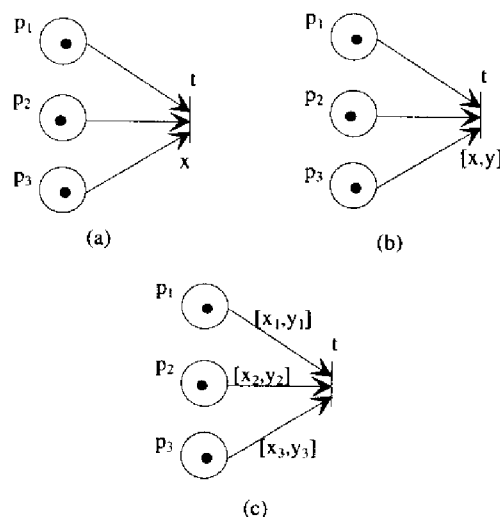


Figure 3-1: Les réseaux de Petri et le temps. (a)réseau de Petri temporisé. (b) réseau de Petri temporel. (c) réseau de Petri à arcs temporels.

Ainsi, pour chacun des modèles abordés on étudiera ses pouvoirs d'analyse, d'expression et de modélisation de scénarios de synchronisation multimédias. En particulier on étudiera la capacité de chacun des modèles à

²Nous ne nous attachons pas dans cette étude à l'évaluation des performances des systèmes multimédias, aussi nous n'aborderons pas les réseaux de Petri stochastiques.

représenter les entités constitutives d'un scénario de synchronisation multimédia, à savoir: les unités de synchronisation, les flux, la synchronisation intra et inter-flux, les intervalles de validité temporelle, la gigue des unités de synchronisation, la dérive inter-flux. On attachera également une grande importance à la capacité qu'ont ces modèles à définir une sémantique formelle satisfaisante pour la notion de synchronisation en environnement faiblement synchrone.

Les apports et limitations des principales extensions temporelles aux réseaux de Petri proposées à ce jour seront concrètement mis en évidence en spécifiant à l'aide de chacun des modèles étudiés un même schéma de synchronisation continue entre un flux vidéo et un flux audio. Cet exemple a été retenu car il synthétise les points clefs de la synchronisation multimédia.

Dans le chapitre 2 nous avons mis en évidence que le paradigme de la régie de production permet une modélisation très simple et naturelle de la synchronisation multimédia, si bien que ce paradigme peut être considéré comme une référence en matière de pouvoir de modélisation. Ainsi de façon à analyser le pouvoir de modélisation de chacun des modèles étudiés, son pouvoir de modélisation sera confronté à celui du « paradigme de la régie de production ». Afin d'évaluer le pouvoir d'analyse de chacun des modèles étudiés les techniques d'analyse qu'ils offrent seront également considérées.

6.2 Les Réseaux de Petri Temporisés

6.2.1 Introduction

Les premiers modèles de réseaux de Petri à extensions temporelles ont été définis afin de permettre la description de systèmes dans lesquels les valeurs explicites du temps possèdent un rôle primordial. Ainsi, ces modèles ont été obtenus en associant aux transitions ou aux places des valeurs explicites du temps représentant des durées soit liées à l'apparition d'événements ou de résultats, soit à la présence de conditions ou d'états. Ainsi les réseaux de Petri temporisés associent un délai à chaque transition [Ram74] ou de façon équivalente à chaque place [Shi77] (Figure 3-1-a). Cette modélisation a été surtout développée dans un but d'évaluation de performances, et d'ordonnement de tâches. Les méthodes d'analyse développées sur les réseaux de Petri temporisés portent essentiellement sur des réseaux saufs, vivants et persistants, ou bien sur des réseaux décomposables en machines à états. Ces méthodes permettent principalement de déterminer les performances d'un système (temps de réponse, débit, temps de cycle minimum) ainsi que l'ordonnançabilité d'un ensemble de tâches [Bra83] [Shi79].

6.2.2 Définitions Formelles

Définition 3.1: Un réseau de Petri temporisé est un doublet $R'=(R,délai)$ avec:

- $R=(P,T,Pré,Post,M_0)$ un réseau de Petri
- $DP: P \rightarrow Q^+ \cup \{+\infty\}$ une application associant un délai à chaque place.

De plus on suppose qu'une horloge permettant de faire correspondre à toute date UTC t une date de simulation $H(t)$ est associée au réseau temporisé.

Les règles de tir d'une transition t d'un réseau de Petri temporisé sont définies de la façon suivante:

Une transition t est tirable à l'instant τ si et seulement si les conditions suivantes sont vérifiées:

- la transition est sensibilisée
- $\forall p_i \in {}^*t$, si τ_i désigne la date de marquage de p_i , alors $\tau = \max_i(\tau_i + DP(p_i))$

Le tir d'une transition est instantané et induit une évolution traditionnelle du marquage du réseau, à savoir:

$$\forall p \in P, M'(p) = M(p) + \text{Post}(p, t) - \text{Pré}(p, t)$$

La notion d'état d'un réseau de Petri temporisé introduite dans [MI80] (ou de façon légèrement différente dans [Zub80]) sous la forme d'un couple comprenant le marquage courant et l'ensemble des temps résiduels associés aux transitions sensibilisées, permet de réaliser une simulation formelle du comportement dynamique d'un système et de générer un graphe d'accessibilité (ou système de transition étiqueté sous-jacent) permettant de procéder à une analyse exhaustive (si le réseau est borné) des propriétés logiques et temporelles du système.

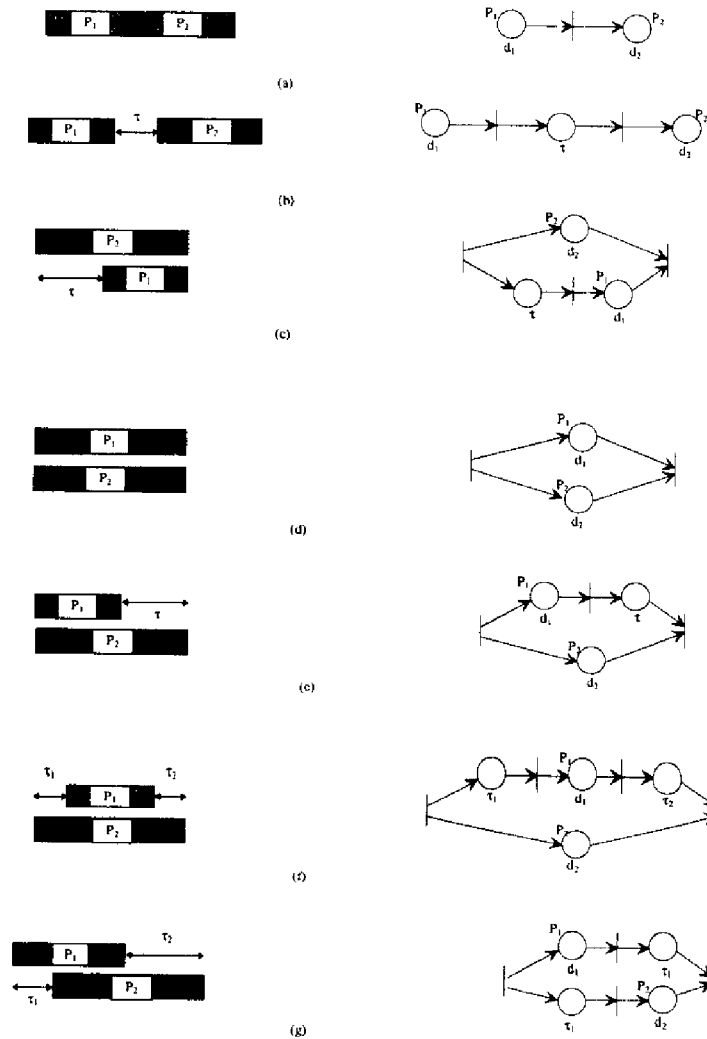


Figure 3-2: Les 7 relations temporelles fondamentales de l'algèbre des intervalles de Allen et leur traduction sous la forme d'un réseau de Petri temporisé. (a) P_1 puis P_2 , (b) P_1 avant P_2 , (c) P_1 finit avec P_2 , $\tau + d_1 = d_2$. (d) P_1 égale P_2 , $d_1 = d_2$. (e) P_1 commence avec P_2 , $d_1 + \tau = d_2$. (f) P_1 pendant P_2 , $d_2 = \tau_1 + d_1 + \tau_2$. (g) P_1 recouvre P_2 , $d_1 + \tau_1 = \tau_2 + d_2$.

6.2.3 Réseaux de Petri Temporisés et Modélisation de la Synchronisation Multimédia

6.2.3.1 Apports du Modèle

Les réseaux de Petri temporisés ont été utilisés par Little et Ghafor [LG90a], sous la dénomination « Object Composition Petri Net » (OCPN) pour la modélisation de scénarios de synchronisation multimédias. Dans le modèle OCPN les places temporisées représentent des unités de synchronisation³ autrement appelées ressources multimédias et sont ainsi typées en fonction du type du média sous-jacent. La principale contribution du modèle OCPN réside dans son approche minimaliste et structurante pour la spécification de scénarios de synchronisation. En effet, un OCPN est construit exclusivement à partir de 13 structures de synchronisation de base représentant les 13 opérateurs de synchronisation fondamentaux de l'algèbre d'intervalles de Allen [All83]. Ces 13 opérateurs temporels peuvent être résumés par 7 relations temporelles de base et leurs relations inverses⁴ (Figure 3.2). Cette approche structurante induit des réseaux de Petri temporisés satisfaisant de « bonnes propriétés ». En particulier, les réseaux de Petri ainsi générés ont une structure de graphe marqué, ils sont « saufs », « conservatifs » et L1-vivants.

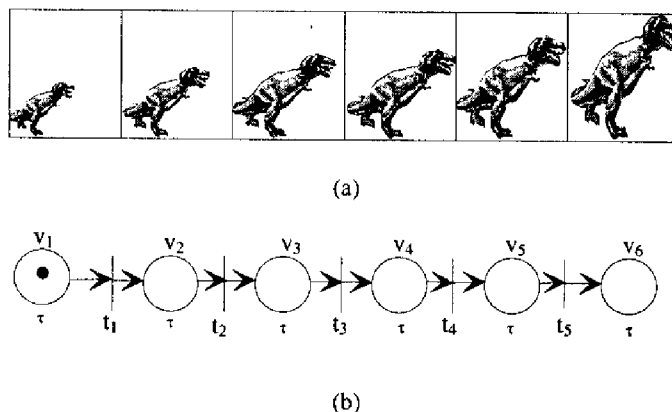


Figure 3-3: Modélisation d'un flux à l'aide des réseaux de Petri temporisés. (a) La séquence d'unités d'information associées à un flux vidéo. (b) Le réseau de Petri temporisé correspondant. Chaque unité de synchronisation, modélisée sous la forme d'une place temporisée à l'aide d'un délai τ , correspond au traitement d'une unité de synchronisation.

Bien que [LG90a] démontre que des scénarios de synchronisation de complexité arbitraire puissent être spécifiés à partir de ces 13 relations temporelles il apparaît qu'elles ne permettent pas de spécifier des schémas de synchronisation non-déterministes. Or, la possibilité de spécifier de façon incomplète ou indéterminée s'avère fondamentale lors du processus de spécification, car elle est la base de tout processus de raffinement de spécifications permettant de faire évoluer des spécifications générales vers des spécifications plus spécialisées. De plus, le caractère indéterminé d'une spécification induit un relâchement des contraintes de conception et une extension des options de conception compatibles avec les spécifications.

³Dans le modèle OCPN des places temporisées peuvent représenter également de purs délais temporels.

⁴La relation de parallélisme strict (Figure 3.2 d) étant symétrique elle est égale par définition à son inverse.

Nous avons vu dans la section 3 que la complétude du pouvoir d'expression d'un modèle pour la synchronisation multimédia basé sur des relations temporelles entre intervalles⁵ se juge à sa capacité à exprimer au moins les 29 relations temporelles indéfinies entre intervalles. Bien que les 13 opérateurs fondamentaux de l'algèbre des intervalles de Allen appartiennent à cet ensemble minimal de relations temporelles ils ne suffisent pas à eux seuls pour assurer cette condition nécessaire de complétude.

Cependant il apparaît que le formalisme des réseaux de Petri temporisés permet intrinsèquement d'exprimer un sur-ensemble de ces 29 relations temporelles indéfinies [WR94]. Ainsi le formalisme des réseaux de Petri temporisés possède le pouvoir d'expression minimum qu'on est en droit d'attendre d'une technique de spécification de scénario de synchronisation multimédia. Cependant, nous verrons dans la section suivante que le pouvoir d'expression des réseaux de Petri temporisés n'est pas suffisant pour permettre de spécifier de façon complète des scénarios de synchronisation multimédias.

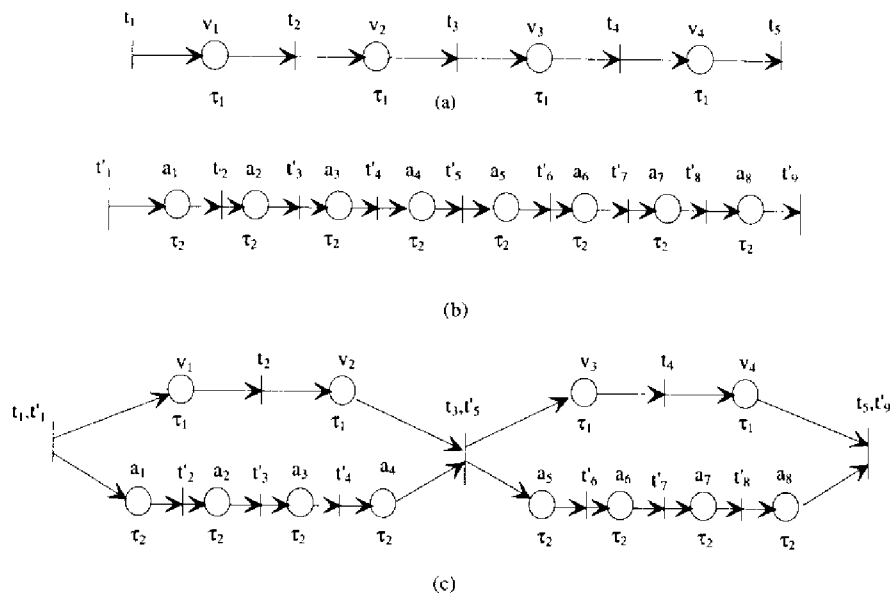


Figure 3-4: Un schéma de synchronisation inter-flux modélisé à l'aide d'un réseau de Petri temporisé. (a) Un flux vidéo composé de 4 unités de synchronisation. (b) Un flux audio composé de 8 unités de synchronisation. (c) Un schéma de synchronisation inter-flux obtenu par fusion de transitions entre les deux flux.

Les travaux de Little et Ghafoor ont également mis en évidence que le pouvoir de modélisation des RdP temporisés permet de reproduire aisément le paradigme de la régie de production. Ainsi conformément à la Figure 3-3 un flux ou schéma de synchronisation intra-flux trouve une correspondance immédiate sous la forme d'une séquence de places temporisées (*i.e.* une séquence d'unités de synchronisation). Le comportement

⁵Les réseaux de Petri Temporels ne permettent de définir des relations de synchronisation qu'entre les événements associés aux dates de terminaison d'intervalles. La notion d'intervalle apparaît pas explicitement dans ce modèle.

dynamique d'un flux peut être alors formellement simulé à partir des règles de tir déterminant l'évolution des jetons à travers le réseau de Petri temporisé.

Les possibilités de modélisation hiérarchiques offertes par les Réseaux de Petri permettent de remplacer dans un flux une séquence de places (*i.e.* un sous-flux) par une place abstraite équivalente et ayant un délai égal à la somme des délais des places composant le sous-flux considéré. Cette possibilité de hiérarchisation offerte par les réseaux de Petri temporisés permet de moduler la granularité des schémas de synchronisation intra-flux conformément au concept d'unité de synchronisation. De même, les schémas de synchronisation inter-flux peuvent être facilement spécifiés par simple fusion de transitions entre plusieurs flux. La Figure 3-4 présente un schéma de synchronisation inter-flux mettant en jeu un flux vidéo et un flux audio. Cet exemple illustre la capacité qu'ont les réseaux de Petri temporisés à reproduire le paradigme de la régie de production pour la spécification de schémas de synchronisation inter-flux. En effet les schémas de synchronisation inter-flux sont obtenus simplement en « superposant » plusieurs flux et en fusionnant certaines de leurs transitions.

Dans un schéma de synchronisation continue tel que celui illustré par la figure 3-4 la granularité de synchronisation inter-flux est ainsi déterminée par la fréquence des points de synchronisation introduits à l'aide de fusions de transition. La notion de synchronisation continue oblige à introduire des points de synchronisation entre des sous-flux de mêmes durées temporelles. Dans l'exemple considéré, les points de synchronisation sont introduits de façon continue entre le flux vidéo et le flux audio dans un rapport 2:4, c'est à dire que deux unités de synchronisation vidéo successives sont synchronisées avec quatre unités de synchronisation audio correspondantes (le schéma de synchronisation inter-flux illustré par la Figure 3-4.c est donc reproduit jusqu'à concurrence de la durée des flux audio et vidéo). La granularité de synchronisation inter-flux est ainsi fonction de la fréquence des points de synchronisation établis entre les flux considérés.

6.2.3.2 Limitations du Modèle

La principale limitation des réseaux de Petri temporisés réside dans leur incapacité à exprimer la notion d'intervalle. En effet ce modèle substitue à la notion d'intervalle la durée ou amplitude de l'intervalle. Cette abstraction de la notion d'intervalle introduit les deux limitations rédhitoires suivantes:

- les réseaux de Petri temporisés ne permettent pas d'exprimer l'indéterminisme temporel associé à une unité de synchronisation. En effet, nous avons vu dans le Chapitre 2 que la variabilité des unités de synchronisation ou gigue s'exprime à l'aide d'intervalles de validité temporelle. Ainsi, en associant une durée précise aux unités de synchronisation modélisées sous forme de places temporisées, ce modèle ne permet pas d'exprimer la gigue des unités de synchronisation, et par là même la notion de dérive inter-flux. Le pouvoir d'expression des réseaux de Petri temporisés n'est donc pas suffisant pour permettre la spécification de la variabilité temporelle car il ne permet de spécifier qu'un seul point dans un intervalle de validité temporelle: la durée nominale de l'unité de synchronisation.
- nous avons vu précédemment que la possibilité de spécifier de façon incomplète est un trait fondamental pour une méthode de spécification. Or, les réseaux de Petri temporisés, par définition, ne permettent pas de spécification partielle ou indéfinie des durées des unités de synchronisation. Seule la notion d'intervalle offre

au spécifieur toute latitude dans l'indéterminisme temporel qu'il peut associer à la spécification de la durée d'une unité de synchronisation.

En résumé, les réseaux de Petri temporisés ont un pouvoir de modélisation permettant de spécifier aisément des scénarios de synchronisation multimédias. Ce pouvoir de modélisation est renforcé par l'utilisation des structures de synchronisation de Allen induisant, de plus, de « bonnes propriétés » aux réseaux résultants, et également susceptibles de faciliter l'analyse de leurs propriétés temporelles. Cependant, ce modèle, en associant des durées exactes aux places, ne permet pas de spécifier la variabilité temporelle intrinsèque aux unités de synchronisation dans les systèmes multimédias faiblement synchrones. En outre, le pouvoir d'expression des réseaux de Petri temporisés ne permet pas de spécifier de façon totalement indéterminée des schémas de synchronisation; en effet ce modèle oblige à spécifier de façon déterministe les durées exactes des unités de synchronisation mises en jeu dans les schémas de synchronisation. Ainsi, bien qu'apportant des contributions notables à la modélisation formelle de la synchronisation dans les systèmes multimédias, les limitations du pouvoir d'expression des réseaux de Petri temporisés ne permettent pas de modéliser de façon complète les contraintes de synchronisation dans les systèmes multimédias. La principale limitation du pouvoir d'expression des réseaux de Petri temporisés, provenant de leur incapacité à modéliser l'indéterminisme temporel, nous conduit à introduire et évaluer dans la section suivante la principale extension temporelle ayant été apportée aux réseaux de Petri afin de combler cette lacune.

6.3 Les Réseaux de Petri Temporels

6.3.1 Introduction

Les réseaux de Petri Temporels ont été introduits par Merlin afin de modéliser la capacité d'un système à se reconfigurer à la suite de pannes transitoires [Mer74]. Ainsi, en associant à chaque transition d'un réseau de Petri classique deux paramètres temporels (*i.e.* un intervalle temporel) dont la sémantique correspond respectivement aux délais minimum et maximum de sensibilisation d'une transition avant qu'elle ne soit tirée, ce modèle permet de spécifier l'indéterminisme temporel des traitements. Plus précisément, si nous considérons, dans un réseau de Petri temporel, une transition t ayant un intervalle temporel associé $[x,y]$, alors tant que cette transition n'est pas sensibilisée ce couple de valeurs est inopérant. Par contre, si la transition est sensibilisée à la date τ alors un compteur virtuel associé à la transition est activé et la transition t est tirable dans tous les instants appartenant à l'intervalle $[\tau+x, \tau+y]$. Cela veut donc dire que la transition t n'est pas tirable avant l'instant $\tau+x$ et qu'elle doit être tirée au plus tard à l'instant $\tau+y$. Ainsi les réseaux de Petri temporisés apparaissent comme un cas particulier de réseaux de Petri temporels.

Merlin utilisait principalement l'intervalle temporel associé à une transition de façon à modéliser par l'intermédiaire de la borne minimum la notion de « chien de garde ». Ce dernier permet de mettre en évidence l'apparition d'un état illégal et, par l'intermédiaire de l'amplitude de l'intervalle (*i.e.* la différence entre la borne maximum et la borne minimum), la durée maximum de reconfiguration du système à partir de l'état illégal détecté. Cependant, d'une façon plus générale, il apparaît que l'intervalle que l'on associe dans un tel réseau à toute transition permet de modéliser l'indéterminisme temporel intrinsèque à tout traitement dans un système faiblement synchrone.

6.3.2 Définitions Formelles

Définition 3.2 : Un réseau de Petri temporel est un doublet $R'=(R, \text{int})$ avec:

- $R=(P, T, \text{Pré}, \text{Post}, M_0)$ un réseau de Petri
- $\text{IT}: T \rightarrow (Q^+ \cup +\infty) \times (Q^+ \cup +\infty)$ une application associant un intervalle temporel à chaque transition telle que, $\forall t_i \in T, \text{IT}(t_i) = (x_i, y_i), (x_i \leq y_i \leq +\infty)$.

L'utilisation d'intervalles rationnels et non réels associés aux transitions est indispensable pour la vérification du caractère borné ou non borné d'un réseau de Petri temporel.

En supposant qu'une horloge, permettant de faire correspondre à toute date UTC τ une date de simulation $H(\tau)$, est associée au réseau temporisé, les règles de tir d'une transition t d'un réseau de Petri temporisé sont définies de la façon suivante:

Une transition t telle que $\text{IT}(t) = (x, y)$ est tirable à l'instant τ si et seulement si les conditions suivantes sont vérifiées:

- la transition est sensibilisée, soit s sa date de sensibilisation.
- $\tau \in [s+x, s+y]$, la transition t doit être tirée au plus tard à l'instant $s+y$.

Lors du tir d'une transition t , le marquage évolue de façon traditionnelle, à savoir:

$$\forall p \in P, M'(p) = M(p) + \text{Post}(p, t) - \text{Pré}(p, t)$$

Une sémantique formelle des réseaux de Petri temporels est introduite dans [BD91]. En particulier la notion d'état d'un réseau de Petri temporel, introduite dans [BD91] comme étant un couple constitué du marquage courant du réseau et de l'ensemble des intervalles temporels dynamiques⁶ associés aux transitions sensibilisées, permet de simuler formellement l'évolution temporelle d'un réseau de Petri temporel. [BD91] introduit également, à partir de la notion de classe d'états, une méthode énumérative permettant, si le réseau de Petri sous-jacent est borné, une évaluation exhaustive du comportement d'un réseau de Petri temporel. Plus généralement, les travaux de [BD91] ainsi que ceux décrits dans [BAB93] posent les bases formelles de méthodes d'analyse de réseaux de Petri permettant de spécifier la variabilité temporelle des événements.

6.3.3 Réseaux de Petri Temporels et Modélisation de la Synchronisation Multimédia

6.3.3.1 Apports du Modèle

Les réseaux de Petri temporels permettent de simuler les arcs inhibiteurs: ils ont donc le pouvoir d'expression d'une machine de Turing [Pet81]. Le modèle proposé par Merlin offre donc un pouvoir d'expression maximal. La contribution de ce modèle à l'étude de la synchronisation dans les systèmes multimédias dépend donc principalement de sa capacité à exprimer aisément les entités constitutives d'un scénario multimédia. Les unités de synchronisation multimédias trouvent une traduction directe dans ce modèle sous la forme d'une place et

⁶La fonction IT associe à chaque transition un intervalle temporel statique. Durant la sensibilisation d'une transition cet intervalle statique se transforme en intervalle dynamique c'est à dire en intervalle dont on fait évoluer les bornes vers zéro conformément à la progression du temps. Ainsi, dans un état donné, l'intervalle dynamique associé à chaque transition sensibilisée spécifie les durées relatives minimum et maximum de sensibilisation de chaque transition à partir de la date d'apparition de l'état considéré.

d'une transition temporisée connectées par l'intermédiaire d'un arc (figure 3-5.a). Dans une telle structure de synchronisation élémentaire, également appelée unité de synchronisation, la place représente une tâche et la transition temporisée associée représente les contraintes temporelles associées à la tâche, ou la fonction de contrôle forçant le respect des contraintes temporelles. Ainsi, un schéma de synchronisation intra-flux s'exprime aisément sous la forme d'une séquence de telles structures de synchronisation (figure 3-5.b). La notion d'intervalle de validité temporelle trouve son expression formelle à travers l'intervalle temporel associé à chaque transition. Tout flux d'un réseau de Petri temporel peut être réduit sous la forme d'une unité de synchronisation dont l'intervalle temporel est la somme⁷ des intervalles temporels des transitions constitutives du flux. Inversement, toute unité de synchronisation d'un réseau de Petri temporel peut être remplacée par un flux (*i.e.* une séquence d'unités de synchronisation) dont la somme des intervalles temporels est égale à l'intervalle temporel de l'unité de synchronisation considérée. Cette caractéristique permet d'exprimer dans le cadre formel des réseaux de Petri temporels la notion de granularité de schémas de synchronisation intra-flux.

6.3.3.2 Limitations du Modèle

Les réseaux de Petri temporels permettent d'exprimer pleinement la variabilité temporelle des unités de synchronisation à l'aide des intervalles temporels associés. Il convient de noter cependant que pour permettre une expression complète de la notion d'intervalle de validité temporelle ainsi que de la gigue admissible d'une unité de synchronisation il serait nécessaire d'associer à chaque transition non plus un doublet temporel mais un triplet temporel comprenant la durée nominale de l'unité de synchronisation considérée. La principale limitation présentée par ce modèle provient du fait que les réseaux de Petri temporels ne permettent pas d'exprimer aisément, c'est à dire conformément au paradigme de composition de la « régie de production », les schémas de synchronisation inter-flux. En effet, nous avons vu précédemment que la fusion de transitions de flux modélisés selon le formalisme des réseaux de Petri, offre une traduction formelle et conforme au paradigme de la « régie de production » des notions de synchronisation inter-flux et de granularité de synchronisation inter-flux. Or suite à la fusion de deux transitions temporisées se pose le problème de la détermination de l'intervalle temporel qui doit être associé à la transition résultante. Etant donné qu'un ensemble de transitions temporisées est remplacé par une seule transition temporisée, il apparaît immédiatement que cette fusion de transitions ne permet pas de conserver les intervalles temporels des diverses unités de synchronisation participant à ce point de synchronisation.

En effet, suite à une analyse pessimiste⁸ des contraintes de synchronisation du schéma de synchronisation inter-flux explicité par la Figure 3-5.d, on constate qu'il est impossible de trouver un intervalle temporel $[x,y]$, associé au point de synchronisation inter-flux modélisé par la transition t_w , satisfaisant dans tous les cas les contraintes temporelles des deux unités de synchronisation impliquées dans ce point de synchronisation (il en est ainsi par exemple si l'on prend $[x_1,y_1]=[1,2]$ et $[x_2,y_2]=[2,3]$).

⁷La somme de deux intervalles $[x_1,y_1]$ et $[x_2,y_2]$ est définie comme étant l'intervalle $[x_1+x_2,y_1+y_2]$.

⁸Une analyse pessimiste de contraintes de synchronisation inter-flux est obtenue en considérant que toutes les unités de synchronisation d'un flux terminent à leur date au plus tôt (*i.e.* la borne inférieure de leur intervalle de validité temporelle) et que inversement toutes les unités de synchronisation de l'autre flux terminent au plus tard (*i.e.* la borne supérieure de leur intervalle temporel).

Ainsi, en associant un intervalle aux transitions d'un réseau de Petri ce modèle ne permet pas de spécifier que les diverses unités de synchronisation impliquées dans un point de synchronisation évoluent de façon asynchrone et ont chacune des contraintes temporelles spécifiques. Autrement dit, les réseaux de Petri temporels induisent un synchronisme artificiel entre les unités de synchronisation impliquées dans un schéma de synchronisation inter-flux. Ce synchronisme explicite à travers l'intervalle unique partagé par l'ensemble des unités de synchronisation considérées, induit, par l'effet des règles de tir des réseaux de Petri temporels, des schémas de synchronisation inter-flux toujours conduits par le flux le plus en retard.

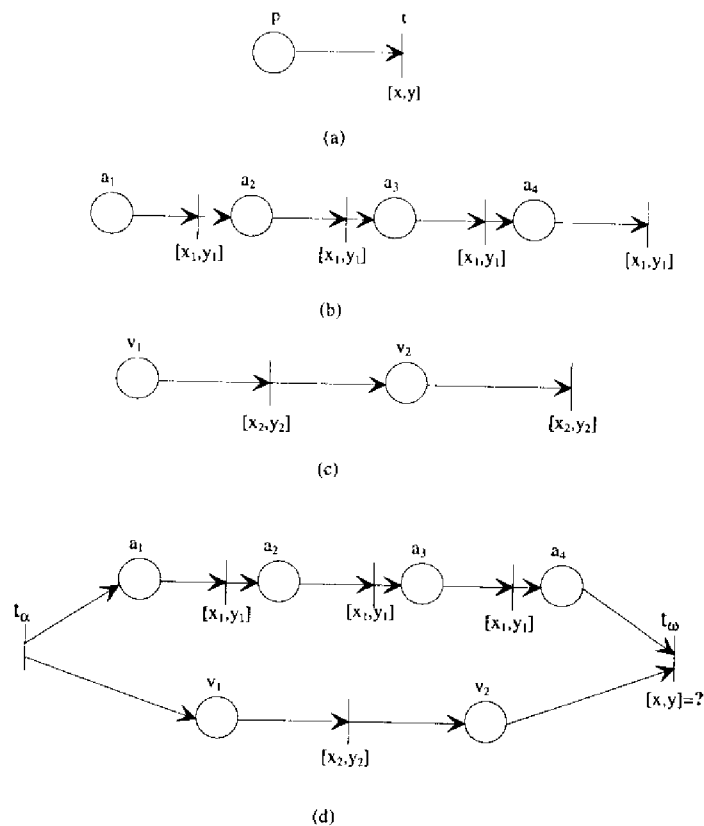


Figure 3-5: Modélisation de la synchronisation intra et inter-flux à l'aide des réseaux de Petri temporels. (a) Modélisation d'une unité de synchronisation. (b) et (c) Modélisation respectivement d'un flux audio et d'un flux vidéo. (d) Modélisation d'un schéma de synchronisation inter-flux.

Ainsi, bien que possédant un pouvoir d'expression et d'analyse pleinement satisfaisants, les limitations du pouvoir de modélisation des réseaux de Petri temporels ne permettent pas de retenir ce modèle comme technique de spécification formelle de scénarios de synchronisation multimédia.

6.4 Les Réseaux de Petri à Arcs Temporels

6.4.1 Introduction

Nous avons vu que le pouvoir d'expression des réseaux de Petri temporisés se trouve renforcé par l'utilisation d'intervalles temporels associés aux transitions. Cependant l'association d'intervalles temporels aux événements (*i.e.* aux transitions) ne permet pas de spécifier aisément des schémas de synchronisation mettant en jeux des traitements évoluant de façon asynchrone. D'autre part l'association d'intervalles temporels aux conditions (*i.e.* aux places) ne permettrait pas de modéliser l'indéterminisme temporel d'un traitement. En particulier, une telle solution ne permettrait pas de modéliser un « time-out » associé à un traitement temporisé. D'où l'idée, exploitée dans [Wal83], d'associer des intervalles temporels non plus aux transitions ou aux places mais aux arcs sortant des places (Figure 3-1.c); cette approche est à l'origine d'un nouveau modèle appelé Réseaux de Petri à Arcs Temporels (RdPAT). Dans un RdPAT l'intervalle temporel associé à un arc spécifie les contraintes temporelles d'une tâche. Dans le cas où une transition a un préfixe se limitant à une seule place, la règle de tir de cette transition est identique à celle d'un réseau de Petri temporel en substituant l'intervalle associé à la transition par celui associé à l'arc reliant la place et la transition considérées. La différence apparaît lorsqu'on considère une transition ayant un préfixe d'une cardinalité strictement supérieure à un. Dans ce cas les règles de tir des RdPAT imposent comme précondition au tir de cette transition que l'ensemble des contraintes temporelles associées aux arcs aboutissant à la transition considérée soient simultanément satisfaites. Ainsi, de façon plus formelle, si $(a_i)_{i \in I}$ est l'ensemble des arcs ayant pour extrémité une transition t et si $[x_i, y_i]$ est l'intervalle temporel associé à l'arc a_i sensibilisé à la date τ_i alors la transition t doit être tirée à une date τ appartenant à $\bigcap_{i \in I} [\tau_i + x_i, \tau_i + y_i]$.

6.4.2 Définitions Formelles

Définition 3.3: Un réseau de Petri à arcs temporels est un doublet $R'=(R, \text{aint})$ avec:

- $R=(P, T, \text{Pré}, \text{Post}, M_0)$ un réseau de Petri
- $IA: A^+ \rightarrow (Q^+ \cup +\infty) \times (Q^+ \cup +\infty)$ avec $A^+ = \{(p, t) \in P \times T / \text{Pré}(p, t) \neq 0\}$. Avec AI une application associant un intervalle temporel à chaque arc sortant d'une place, telle que:

$$\forall a_i \in A^+, IA(a_i) = (x_i, y_i) \text{ avec } (x_i \leq y_i)$$

On supposera qu'une horloge permettant de faire correspondre à toute date UTC τ une date de simulation $H(\tau)$ est associée à tout RdPAT. Les règles de tir d'une transition t d'un réseau d'un RdPAT sont alors définies de la façon suivante (en supposant par souci de simplification qu'un arc ne peut être sensibilisé plus d'une fois simultanément):

Une transition t_k d'un RdPAT est tirable à l'instant τ si et seulement si les conditions suivantes sont vérifiées :

- la transition t_k est sensibilisée,
- $\tau \in \bigcap_{i \in I_k} [\tau_i + x_i, \tau_i + y_i]$, avec $I_k = \{i \in N / a_i \in A_k^+\}$, $A_k^+ = \{(p, t_k) \in P \times T / \text{Pré}(p, t_k) \neq 0\}$, et τ_i la date de sensibilisation de l'arc a_i , c'est à dire la date de marquage de la place associée.

Ces règles de tir font intervenir des intervalles temporels absolus (*i.e.* faisant intervenir des dates absolues). Ces intervalles doivent être distingués des intervalles temporels associés aux arcs par la fonction IA, qui sont des intervalles temporels faisant intervenir des instants relatifs à la date de marquage de l'arc associé.

Lors du tir d'une transition t , le marquage évolue de façon traditionnelle, à savoir:

$$\forall p \in P, M'(p) = M(p) + Post(p, t) - Pré(p, t)$$

Les règles de tir des RdPAT induisent implicitement la notion d'*incohérence temporelle* ou *désynchronisation*. En effet, une incohérence temporelle survient lorsque l'intersection des intervalles absolus associés aux arcs d'une transition sensibilisée est vide. Dans ce cas, conformément aux règles de tir précédemment énoncées, la transition considérée ne peut pas être tirée. Le comportement d'un RdPAT est alors indéfini [Wal83]. Les incohérences temporelles ne sont pas admises dans un RdPAT, à ce titre elles sont considérées comme des erreurs de spécification. Ainsi, il est fondamental de disposer de techniques d'analyse permettant de mettre en évidence les risques d'incohérence temporelle dans un RdPAT. Cependant aucun algorithme⁹ n'a été proposé par Walter pour déterminer de façon efficace les risques de désynchronisation dans un RdPAT.

D'une façon différente [Bol90i] exploite le modèle de synchronisation prôné par les RdPAT à partir d'une algèbre de processus. Les problèmes de désynchronisation sont supprimés (mais non résolus) en étendant la sémantique de synchronisation de façon telle qu'en cas d'incohérence temporelle les jetons mis en cause sont retirés du réseau de Petri.

6.4.3 Réseaux de Petri à Arcs Temporels et Modélisation de la Synchronisation Multimédia

6.4.3.1 Apports du Modèle

Le principal avantage des RdPAT pour la modélisation de la synchronisation multimédia réside dans leur pouvoir de modélisation. En effet ce modèle permet une modélisation simple, et très proche du paradigme de la régie du production, des entités fondamentales constitutives des scénarios de synchronisation multimédias. Ainsi, à l'aide d'un RdPAT, une unité de synchronisation est modélisée sous la forme d'un arc temporel (Figure 3.6-a). Un flux s'exprime alors simplement sous la forme d'une séquence d'arcs temporels (Figures 3.6-b et 3.6-c). L'intervalle temporel associé à chaque arc sortant permet d'exprimer la variabilité temporelle des unités de synchronisation. Le modèle RdPAT permet également d'exploiter le concept de granularité de synchronisation intra-flux à l'aide « d'arcs abstraits » représentant sous une forme condensée une séquence d'arcs. L'intervalle temporel associé à un arc abstrait est obtenu par sommation des intervalles associés à la séquence des places sous-jacentes. Inversement, tout arc temporisé peut être substitué par une séquence d'arcs temporisés dont la somme des intervalles est égale à l'intervalle associé à l'arc objet de la substitution. Les schémas de synchronisation inter-flux sont aisément modélisables par simple fusion de transitions appartenant aux flux devant être synchronisés. Ainsi, ce modèle permet d'exprimer la variabilité temporelle des traitements tout en permettant, à la différence des réseaux de Petri de Merlin, une expression aisée des schémas de synchronisation inter-flux. La granularité de

⁹ [Walter] met en évidence que de tels algorithmes sont, dans le cas général, de complexité non-polynomiale.

synchronisation inter-flux s'exprime naturellement à l'aide de points de synchronisation (résultant de fusions de transitions) placés de façon plus ou moins fréquente entre plusieurs flux.

6.4.3.2 Limitations du Modèle

Bien que doté d'un pouvoir de modélisation satisfaisant, les pouvoirs d'expression et d'analyse du modèle RdPAT n'en présentent pas moins des limitations rédhibitoires. Le principal désavantage associé à l'utilisation des RdPAT résulte de leur incapacité à résoudre de façon satisfaisante les cas de désynchronisation inter-flux. Cette limitation est d'autant plus pénalisante que dans un système faiblement synchrone, ou a fortiori asynchrone, les risques de désynchronisation sont potentiellement inhérents à tout schéma de synchronisation inter-flux. En effet, dans certains schémas de synchronisation inter-flux la dérive inter-flux peut induire des incohérences temporelles. Il en est ainsi par exemple si l'on prend dans le schéma de synchronisation illustré par la Figure 3-6.d les valeurs suivantes pour les intervalles temporels associés respectivement aux unités de synchronisation audio et vidéo, $[x_1, y_1]=[1,3]$, $[x_2, y_2]=[3,5]$. On supposera de plus que les durées nominales des unités de synchronisation audio et vidéo sont respectivement égales à 2 et 4 unités de temps. Ainsi en l'absence de gigue ce schéma de synchronisation inter-flux permet de respecter simultanément les contraintes de synchronisation intra-flux des deux flux. Cependant si l'on intègre la gigue admissible des unités de synchronisation, une analyse pessimiste de ce schéma de synchronisation (*i.e.* les unités de synchronisation audio se terminent à leur date au plus tard, soit une durée de 3 unités de temps, alors que les unités de synchronisation vidéo se terminent à leur date au plus tôt, soit une durée de 3 unités de temps) montre que ce schéma de synchronisation inter-flux est exposé à un risque d'incohérence temporelle. En effet, dans le cas le plus pessimiste l'intervalle temporel absolu associé à l'arc (a_4, t_w) est égal à $[3+3+3+1, 3+3+3+3] = [10,12]$; de même l'intervalle temporel associé à l'arc (v_2, t_w) est égal à $[3+3, 3+5]=[6,8]$. Ainsi dans ce cas l'intersection des intervalles temporels des unités de synchronisation associées à ce point de synchronisation est vide. Dans un tel cas, les règles de synchronisation prescrites par Walter stipulent un comportement indéterminé. Aussi ce schéma de synchronisation inter-flux devrait-il être considéré comme une erreur de conception et rejeté suite à une analyse du réseau. Il paraît très excessif de rejeter un schéma de synchronisation qui, dans le cas nominal et dans la plupart des cas (*i.e.* lorsque les ressources sont suffisamment disponibles), est cohérent mais qui dans des conditions limites est exposé à des problèmes de désynchronisation. Ainsi serait-il plus réaliste de disposer d'une sémantique de synchronisation de type « best effort » proposant un compromis entre le rejet systématique des schémas de synchronisation « à risque » (qui ne sont pas toujours décelables avec une complexité de calcul polynomiale) et l'admission sans distinction de tout schéma de synchronisation, en s'exposant à des incohérences temporelles entraînant un comportement applicatif indéterminé.

Les règles de tir prônées par [Bol90], bien qu'induisant une sémantique de synchronisation plus robuste face aux problèmes de désynchronisation, ne sont pas pour autant satisfaisantes pour résoudre la problématique de la synchronisation multimédia en environnement faiblement synchrone. En effet en prescrivant le retrait des jetons en cas de désynchronisation ces règles de tir ne sont pas capables d'assurer la continuité des flux. Ainsi dans l'exemple de synchronisation inter-flux illustré par la Figure 3-6.d, si un problème de désynchronisation survient en un point quelconque de synchronisation tel que celui modélisé par la transition t_w , alors les flux vidéo et audio sont interrompus. Or, nous avons vu dans le Chapitre 2 que l'on peut traiter des problèmes de désynchronisation

dans les flux multimédias en mettant en oeuvre des sauts ou duplications d'unités de synchronisation afin d'assurer la continuité des flux. Autrement dit la permissivité des flux en matière de respect des contraintes de synchronisation milite en faveur de sémantiques de synchronisation faisant primer la continuité des flux sur le respect absolu des contraintes de synchronisation. Ainsi, au regard de la souplesse de synchronisation offerte par les flux multimédias, les règles de synchronisation spécifiées par [Bol90] paraissent trop rigides et trop restrictives.

En terme de limitation du pouvoir d'expression des RdPAT notons de plus que, comme dans le cas des réseaux de Merlin, il conviendrait de rajouter la durée nominale aux intervalles temporels associés aux arcs afin de permettre d'exprimer pleinement la notion d'intervalle de validité temporelle.

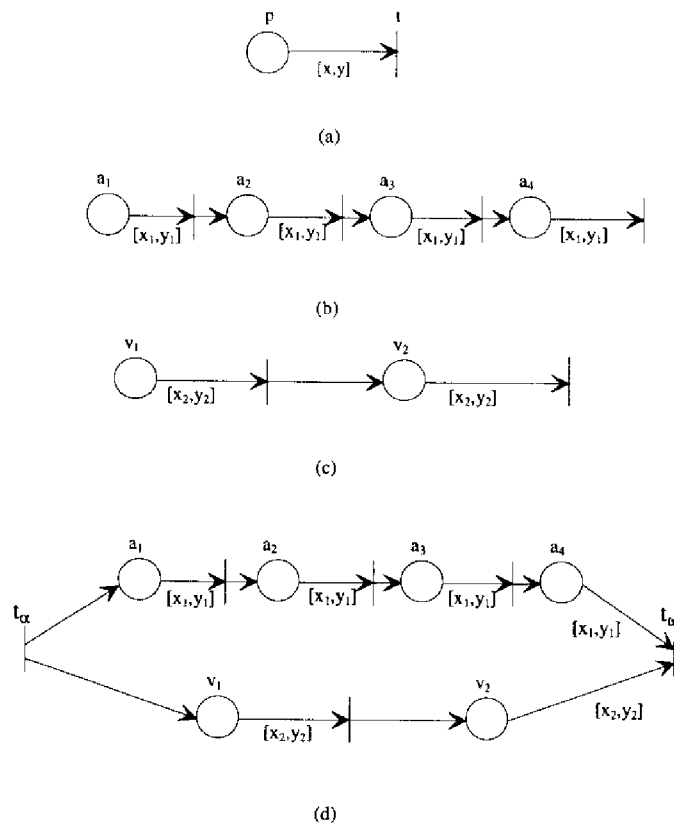


Figure 3-6: Modélisation de la synchronisation intra et inter-flux à l'aide des réseaux de Petri à arcs temporels. (a) Modélisation d'une unité de synchronisation, (b) et (c) Modélisation respectivement d'un flux audio et d'un flux vidéo, (d) Modélisation d'un schéma de synchronisation inter-flux

De plus, à notre connaissance, il n'a été proposé aucune sémantique formelle pour les RdPAT. En particulier il n'existe aucune méthode permettant de réaliser une simulation formelle de l'évolution de l'état d'un système modélisé sous forme de RdPAT. De même aucune méthode d'analyse permettant de générer le système de transition étiqueté sous-jacent à un RdPAT ne semble avoir été proposée à ce jour. Le modèle RdPAT est donc très lacunaire en matière de techniques d'analyse. Ainsi, bien que possédant un bon pouvoir de modélisation, les

limitations des pouvoirs d'analyse et d'expression du modèle RdPAT ne permettent pas de retenir tel quel ce modèle pour la spécification formelle des contraintes de synchronisation dans les systèmes faiblement synchrones.

7. Conclusion

Partant de la considération que les réseaux de Petri offrent un des cadres formels les plus complets pour l'étude des propriétés temporelles et logiques des systèmes multimédias, nous avons réalisé dans ce chapitre un panorama des principales extensions temporelles qui ont été proposées pour ce formalisme. Ce panorama nous a permis de mettre en évidence qu'aucun des modèles existants n'offre simultanément des pouvoirs de modélisation, d'expression et d'analyse satisfaisants pour la spécification formelle des contraintes de synchronisation dans les systèmes faiblement synchrones. Cette constatation justifie pleinement la création d'un nouveau modèle comblant les limitations des modèles existants. En synthétisant les contributions et les limitations de chacun des modèles examinés dans ce panorama il nous est possible de profiler les caractéristiques de ce nouveau modèle. Ainsi l'élaboration de ce nouveau modèle est susceptible de bénéficier des éléments suivants:

- l'utilisation exclusive des opérateurs de synchronisation de l'Algèbre des intervalles de Allen (transcrits dans le formalisme des réseaux de Petri) confère aux réseaux de Pétri résultants de « bonnes propriétés » structurelles ainsi qu'un bon pouvoir de modélisation si l'on fait abstraction de la variabilité temporelle des traitements.
- les pouvoirs d'expression et d'analyse des réseaux de Petri de Merlin peuvent être considérés comme une référence pour tout nouveau modèle.
- le formalisme des RdPAT, en introduisant des intervalles temporels sur les arcs sortant des transitions, apporte un pouvoir de modélisation très séduisant. Cependant ce modèle apparaît bien trop limité en terme de pouvoirs d'expression et d'analyse.
- la notion d'intervalle temporel introduite par les réseaux de Petri temporels et les RdPAT doit être remplacée par un triplet temporel permettant de spécifier l'intervalle de validité temporelle des unités de synchronisation.

Il se profile derrière ces constatations un nouveau modèle, objet du chapitre suivant, empruntant à la fois aux réseaux de Petri de Merlin pour leur pouvoir d'expression et d'analyse, et aux RdPAT pour leur pouvoir de modélisation.



Chapitre 4

Les Réseaux de Petri à Flux Temporels

1. Introduction

Nous avons mis en évidence dans le chapitre précédent que le formalisme des Réseaux de Petri à Arcs Temporels, grâce à l'association d'intervalles temporels aux arcs issus des places, possède au sein de la famille des réseaux de Petri à extensions temporelles, le pouvoir de modélisation le plus abouti pour la spécification des scénarios de synchronisation multimédia. Cependant la sémantique de synchronisation inter-flux proposée par ce modèle n'est pas en mesure d'assurer, en cas de désynchronisation, la continuité des flux ou la continuation des scénarios de synchronisation. Autrement dit, les RdPAT ne fournissent pas une sémantique de synchronisation réaliste pour les systèmes multimédias. En effet, une sémantique de synchronisation en environnement faiblement synchrone doit permettre une gestion explicite des problèmes de désynchronisation induits par les dérives inter-flux. Il n'en demeure pas moins que le pouvoir de modélisation d'une technique de spécification est à nos yeux un critère fondamental pour assurer le succès d'utilisation et de diffusion d'un modèle. Ainsi, à partir d'un formalisme empruntant le pouvoir de modélisation des RdPAT, nous allons définir dans ce chapitre un nouveau modèle introduisant une sémantique formelle complète et précise de la notion de synchronisation en environnement faiblement synchrone. Par sémantique de synchronisation complète et précise nous entendons une sémantique de synchronisation intégrant explicitement de façon complète et précise les potentialités de désynchronisation inter-flux. Ce nouveau modèle comblera ainsi les lacunes et limitations réhivitoires des modèles existants.

2. Organisation du Chapitre

A partir d'une analyse détaillée de la problématique de la synchronisation dans les systèmes faiblement synchrones, ce chapitre introduit dans un premier temps une sémantique de la notion de synchronisation dans les systèmes faiblement synchrones (section 3). Une nouvelle extension temporelle aux réseaux de Petri, appelée Réseaux de Petri à Flux Temporels (RdPFT), permettant de modéliser simplement et complètement cette problématique est alors introduite (section 4), et sa sémantique formelle est explicitée (section 5). Puis, les

propriétés fondamentales de ce nouveau modèle sont étudiées (section 6), et deux exemples de synthèse illustrent les possibilités offertes par ce nouveau formalisme (section 7).

3. Vers une Nouvelle Sémantique de Synchronisation

3.1 Points de Vue

Toute tentative de modélisation constitue un processus d'abstraction de la réalité. Ainsi, un modèle doit être le reflet pragmatique et objectif d'une certaine problématique réelle. L'observation et la formulation de la problématique considérée est donc un préalable incontournable à la définition d'un modèle. Relativement à la problématique de la synchronisation dans les systèmes faiblement synchrones cette constatation induit deux points de vue, à savoir:

- un point de vue « exécution » (ou comportemental), qui étudie et définit précisément le comportement réel d'un exécutif de synchronisation en environnement faiblement synchrone. Ce point de vue s'attache ainsi à définir l'ordonnancement des unités de synchronisation associées à un scénario de synchronisation.
- un point de vue « modélisation », réalisant à partir d'un modèle formel une abstraction du point de vue exécution, et apportant par là même des capacités d'expression, simulation, et analyse des propriétés logiques et temporelles.

Conformément aux cycles de vie prônés par le génie logiciel, le point de vue modélisation doit être le point de départ de toute implantation; inversement la définition d'un modèle doit être conduite à partir de considérations issues d'un point de vue comportemental. Ainsi, dans la section suivante, nous allons mettre en évidence, à partir d'un point de vue exécution, une sémantique de synchronisation cohérente en environnement faiblement synchrone. Cette sémantique de synchronisation induira sur le point de vue modélisation un nouveau modèle dérivé des RdPAT mais s'en distinguant principalement par de nouvelles règles de tir.

3.2 Synchronisation et Unités de Synchronisation

Nous avons vu dans le chapitre précédent que les treize relations temporelles d'Allen permettent d'exprimer de façon formelle et complète la notion de synchronisation entre deux unités de synchronisation dans un environnement synchrone. Ces structures temporelles synchrones peuvent être naturellement étendues à un environnement faiblement synchrone en substituant la notion d'intervalle de validité temporelle à celle d'intervalle. On remarque alors que si les relations temporelles de type « série » (figures 3.2-a et 3.2-b) conservent leur sémantique de synchronisation, il n'en est plus de même des relations temporelles de type « parallèle » (figures 3.2-c à 3.2-g). En effet considérons à titre d'exemple l'opérateur de synchronisation « égale » spécifiant que deux unités de synchronisation débutent et finissent en même temps. Un tel schéma de synchronisation en environnement synchrone est illustré par la figure 4.1-a, alors que sa traduction en environnement faiblement synchrone est donnée par la figure 4.2-b.

Il apparaît dans la figure 4.1-b que la variabilité temporelle des unités de synchronisation introduit un indéterminisme de synchronisation susceptible de transformer un schéma de synchronisation purement

déterministe en une disjonction de schémas de synchronisation. Ainsi la relation temporelle « US_1 égale US_2 » est transformée en $((US_1 \text{ égale } US_2) \vee (US_1 \text{ commence-avec } US_2)) \vee (US_2 \text{ commence-avec } US_1)$. Des artefacts temporels de même nature peuvent être également observés pour chacun des 5 schémas de synchronisation fondamentaux de type parallèle. Il est donc essentiel d'introduire une sémantique de synchronisation en environnement faiblement synchrone qui assure un comportement déterministe des schémas de synchronisation tout en intégrant la variabilité temporelle des unités de synchronisation.

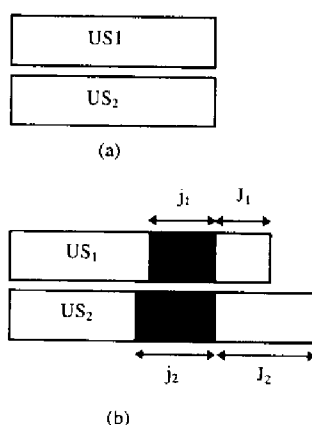


Figure 4.1: L'indéterminisme de synchronisation introduit dans un schéma de synchronisation de type parallèle par la variabilité temporelle des unités de synchronisation. Les unités de synchronisation US_i sont soumises à une gigue positive notée J_i et à une gigue négative notée j_i .

3.3 Instants Fondamentaux

Nous venons de voir que la problématique de la synchronisation en environnement asynchrone nécessite la définition d'une sémantique de synchronisation inter-flux. De façon à introduire formellement cette problématique, considérons le schéma de synchronisation, entre deux unités de synchronisation US_i , d'intervalles de validité temporelle (x_i, n_i, y_i) , avec $i \in \{1, 2\}$, illustré par la figure 4.2. Un tel schéma de synchronisation revêt un caractère générique car il permet de spécifier n'importe lequel des 13 schémas fondamentaux de synchronisation¹. Dans ce schéma générique de synchronisation en environnement faiblement synchrone les arcs (τ_1, t_1) et (τ_2, t_2) représentent de façon abstraite les caractéristiques temporelles de l'ensemble des unités de synchronisation conduisant à l'activation des deux unités de synchronisation considérées. De plus, de façon à permettre une spécification complète de la notion d'intervalle de validité temporelle, l'intervalle temporel associé à chaque arc sortant d'une place a été étendu en lui intégrant la durée nominale de l'unité de synchronisation ainsi modélisée.

¹En supposant que les unités de synchronisation considérées peuvent représenter de purs délais temporels indéterministes.

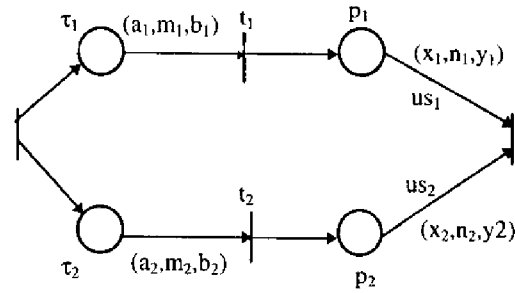


Figure 4.2: Un schéma de synchronisation inter-flux générique

A partir d'un point de vue exécution, six instants fondamentaux de synchronisation peuvent être associés au point de synchronisation inter-flux modélisé par la transition t . Ces 6 instants absolus détaillés dans le Tableau 4-1 caractérisent les événements temporels déterminants pour l'ensemble des unités de synchronisation impliquées dans ce point de synchronisation. Il convient de noter que l'intervalle de validité temporelle $[x_i, n_i, y_i]$ d'une unité de synchronisation US_i , qui est constitué d'instant relatifs, correspond par le biais de ces instants fondamentaux à l'intervalle absolu $[\tau_i^{\min}, \tau_i^{\text{nom}}, \tau_i^{\max}] = [\tau_i^{\text{deb}} + x_i, \tau_i^{\text{deb}} + n_i, \tau_i^{\text{deb}} + y_i]$. Ainsi, on qualifiera $[x_i, n_i, y_i]$ d'intervalle relatif de validité temporelle alors que $[\tau_i^{\min}, \tau_i^{\text{nom}}, \tau_i^{\max}]$ sera appelé intervalle absolu de validité temporelle.

Parmi ces six instants de synchronisation cinq instants peuvent être mis en correspondance avec des événements relatifs au point de vue modélisation. En effet le point de vue modélisation fait abstraction de la date effective de terminaison d'une unité de synchronisation. Ainsi le modèle recouvre à travers la notion d'intervalle de validité temporelle l'ensemble des dates de terminaison potentielles associées à une unité de synchronisation. De même l'intervalle de tir d'une transition (intervalle défini par les règles de tir) spécifie l'ensemble des instants de synchronisation potentiels associés au point de synchronisation considéré.

Ces 6 instants fondamentaux de synchronisation vont nous permettre de spécifier précisément la problématique de la synchronisation en environnement faiblement synchrone et par là même de définir une sémantique formelle de synchronisation levant les limitations des modèles existants.

3.4 Formalisation de la Problématique de Synchronisation en Environnement Faiblement Synchrone

L'objectif d'un exécutif de synchronisation est de faire en sorte que la sémantique temporelle des unités de synchronisation dont il réalise l'ordonnancement soit satisfaite. Ainsi, partant d'un point de vue exécution nous allons étudier et définir les propriétés temporelles qui doivent être garanties pour les schémas de synchronisation intra-flux et inter-flux. Ces propriétés temporelles fourniront alors la base d'une sémantique formelle de la synchronisation en environnement faiblement synchrone.

<u>Instants Fondamentaux</u>	<u>Point de vue Exécution</u>	<u>Point de vue Modélisation</u>
τ_i^{deb}	l'unité de synchronisation u_i débute son exécution.	le jeton marque la place p_i associée à l'arc $a_i=(p_i,t)$ modélisant l'unité de synchronisation u_i .
τ_i^{min}	l'unité de synchronisation atteint la borne inférieure de son intervalle de validité temporelle, notée x_i . Autrement dit, quand l'unité de synchronisation s'exécute depuis x_i unités de temps. Ainsi nous avons $\tau_i^{min}=\tau_i^{deb}+x_i$. L'exécutif de synchronisation doit assurer une durée minimum d'exécution de l'unité de synchronisation égale à x_i .	l'âge du jeton dans la place p_i atteint la borne inférieure x_i de l'intervalle de validité temporelle associé à l'arc $a_i=(p_i,t)$. La durée x_i modélise ainsi la durée d'exécution minimum admissible pour l'unité de synchronisation modélisée par l'arc a_i .
τ_i^{fin}	l'exécution de l'unité de synchronisation u_i se termine effectivement. Idéalement la durée d'exécution de u_i est égale à la durée nominale d'exécution notée n_i . Ainsi, dans le cas nominal, nous avons: $\tau_i^{fin}=\tau_i^{deb}+n_i$. Cependant la sémantique temporelle de l'unité de synchronisation u_i est préservée pour toute les dates de terminaison appartenant à l'intervalle $[\tau_i^{deb}+x_i, \tau_i^{deb}+y_i]$.	Cet instant n'est pas couvert par le point de vue modélisation. Le point de vue modélisation fait abstraction des durées réelles d'exécution des unités de synchronisation.
τ_i^{max}	l'unité de synchronisation atteint la borne supérieure de son intervalle de validité temporelle, notée y_i . Autrement dit, quand l'unité de synchronisation s'exécute depuis y_i unités de temps. Ainsi nous avons $\tau_i^{max}=\tau_i^{deb}+y_i$. Afin de garantir la sémantique temporelle de u_i l'exécutif de synchronisation doit assurer une durée maximum d'exécution de l'unité de synchronisation égale à y_i .	l'âge du jeton dans la place p_i atteint la borne supérieure y_i de l'intervalle de validité temporelle associé à l'arc $a_i=(p_i,t)$. La durée y_i spécifie ainsi la durée d'exécution maximum admissible pour l'unité de synchronisation modélisée par l'arc a_i .
$\tau_i^{tou}=\tau^{tou}$	les unités de synchronisation associées au point de synchronisation considéré ont toutes commencé leur exécution. Cette date est identique pour toutes les unités de synchronisation impliquées dans le schéma de synchronisation inter-flux considéré.	la transition t devient sensibilisée. Autrement dit $\tau_i^{tou} = \max_i(\tau_i^{deb})$
$\tau_i^{syn}=\tau^{syn}$	le point de synchronisation est atteint. Cet instant unique est défini par l'exécutif de synchronisation. Cette date est identique pour toutes les unités de synchronisation impliquées dans le schéma de synchronisation inter-flux considéré.	la transition est tirée. Le point de vue modélisation considère l'ensemble des instants de tir potentiels. Cet ensemble est défini à partir de la notion d'intervalle de tir.

Tableau 4-1: Les instants fondamentaux de synchronisation pour une unité de synchronisation.

3.4.1 Synchronisation Intra-flux

Pour ce qui est des contraintes de synchronisation intra-flux, chaque unité de synchronisation doit satisfaire les contraintes temporelles spécifiées par son intervalle absolu de validité temporelle. Ainsi en l'absence de problème de synchronisation nous avons $\tau_i^{fin} \in [\tau_i^{min}, \tau_i^{max}]$. Un problème de désynchronisation se traduit alors, selon un point de vue exécution, par le fait que $\tau_i^{fin} \notin [\tau_i^{min}, \tau_i^{max}]$. Dans un tel cas l'exécutif de synchronisation doit forcer le respect de la sémantique temporelle de l'unité de synchronisation. L'exécutif réalise cela en garantissant que $\tau^{syn} \in [\tau_i^{min}, \tau_i^{max}]$.

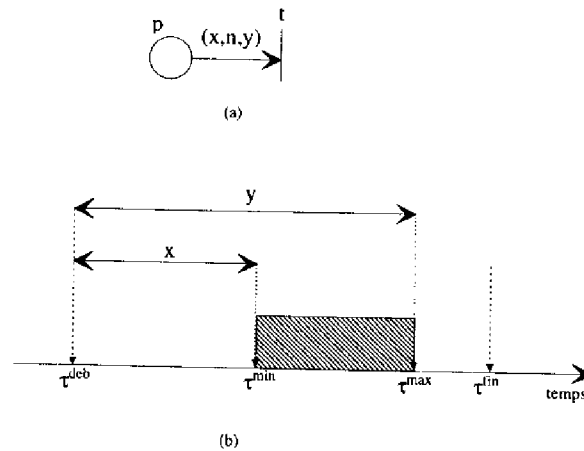


Figure 4.3: Point de vue « exécution » d'un schéma de synchronisation intra-flux

Le rôle de l'exécutif de synchronisation est illustré par la figure 4.3 où il apparaît que si l'unité de synchronisation considérée est en retard, ce qui se traduit par $\tau_i^{\max} < \tau_i^{\text{fin}}$, alors l'exécutif de synchronisation garantit tout de même la sémantique temporelle de l'unité de synchronisation en assurant que $\tau^{\text{syn}} = \tau_i^{\max}$, autrement dit en stoppant l'unité de synchronisation à l'instant τ^{\max} . Une telle stratégie de synchronisation peut être ainsi qualifiée de « best effort ». Inversement si une unité de synchronisation est en avance, ce qui se traduit par $\tau_i^{\text{fin}} < \tau_i^{\min}$, alors l'exécutif de synchronisation doit assurer la sémantique temporelle de l'unité de synchronisation en réalisant $\tau_i^{\text{syn}} = \tau_i^{\min}$. Cette fonction de contrôle réalisée par l'exécutif peut être accompagnée de traitements d'exception conformes aux actions correctives ou de substitution proposées dans [Ste90]. Les actions correctives appliquées par l'exécutif lorsque des problèmes de désynchronisation sont observés ont pour objectif de modifier dynamiquement les paramètres de qualité de service de synchronisation de façon à assurer la cohérence temporelle des futures unités de synchronisation. Les actions de substitution visent quant à elles à assurer pour le mieux la cohérence temporelle d'une unité de synchronisation exposée à un problème de désynchronisation. Ces actions de substitution sont dépendantes de la sémantique applicative et du type de flux auquel appartient l'unité de synchronisation en cause. Par exemple, lorsque une trame vidéo n'est pas disponible, une action de substitution peut consister à présenter à nouveau la dernière trame vidéo disponible.

Les observations concernant la synchronisation intra-flux que nous venons de réaliser d'un point de vue « exécution » induisent selon un point de vue modélisation une sémantique formelle de tir de transition satisfaisant la propriété suivante:

$$\forall a_i \in A^+ / \text{IA}(a_i) = [x_i, y_i], \tau_i^{\text{syn}} \in [\tau_i^{\text{deb}} + x_i, \tau_i^{\text{deb}} + y_i] = [\tau_i^{\min}, \tau_i^{\max}]$$

Autrement dit, l'intervalle temporel associé à un arc (figure 4.3-a) détermine les durées minimum et maximum de sensibilisation de l'arc considéré. On rejoint par là même les règles de tir traditionnelles des RdPAT.

3.4.2 Synchronisation Inter-flux

Considérons un schéma de synchronisation inter-flux entre un ensemble $(us_i)_{i \in I}$ d'unités de synchronisation (figure 4.3). Chacune des unités de synchronisation impliquées dans ce schéma (à savoir $us_1=(P_1,t)$ et $us_2=(P_2,t)$) doit satisfaire ses propres contraintes de synchronisation intra-flux. Ainsi, conformément à la sémantique de synchronisation intra-flux mise en évidence précédemment la propriété suivante doit être satisfaite:

$$\tau^{syn} \in \bigcap_{i \in I} [\tau_i^{min}, \tau_i^{max}] ; \text{ soit } V_i \text{ cet intervalle.}$$

Cependant les deux problèmes de désynchronisation suivants peuvent apparaître dans de tels schémas de synchronisation inter-flux:

1. L'intersection des intervalles absolus de validité temporelle peut être vide, ce qui se traduit par:

$$V_i = \emptyset$$

La figure 4.4 illustre ce cas extrême de désynchronisation pour lequel il est impossible de satisfaire simultanément les contraintes de synchronisation de us_1 et de us_2 . En effet si l'on choisit un instant de synchronisation dans l'intervalle $[\tau_1^{min}, \tau_1^{max}]$ (respectivement $[\tau_2^{min}, \tau_2^{max}]$) alors on respectera les contraintes temporelles de us_1 (respectivement us_2) mais non celles de us_2 (respectivement us_1). De plus il est évident que tout instant de synchronisation n'appartenant pas à $[\tau_1^{min}, \tau_1^{max}] \cup [\tau_2^{min}, \tau_2^{max}]$ ne satisfait ni les contraintes temporelles de us_1 ni celles de us_2 .

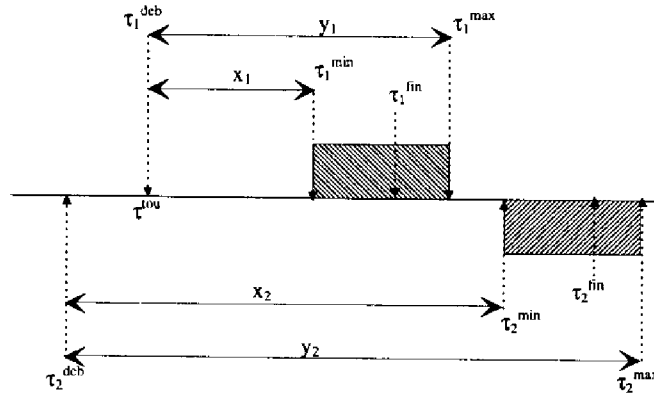


Figure 4.4 : Point de vue « exécution » d'un schéma de synchronisation inter-flux: cas où l'intersection des IAVT est vide.

Si nous appliquons une analyse temporelle pessimiste au schéma générique de synchronisation inter-flux modélisé par la figure 4.2 nous pouvons mettre en évidence qu'un tel problème de désynchronisation peut survenir si et seulement si les inégalités suivantes ne sont pas vérifiées:

$$\begin{aligned} \max_{i \in I} (a_i + x_i) &\leq \min_{j \in I} (b_j + y_j) \\ \max_{i \in I} (b_i + x_i) &\leq \min_{j \in I} (a_j + y_j) \end{aligned} \quad \text{avec } i \neq j$$

ou autrement dit étant donné que $\left(\max_{i \in I} (a_i + x_i) \leq \max_{j \in I} (b_j + x_j) \right) \wedge \left(\min_{i \in I} (a_i + y_i) \leq \min_{j \in I} (b_j + y_j) \right)$, une condition nécessaire et suffisante pour qu'il n'existe pas de risque de désynchronisation est donné par la relation suivante:

$$\max_{i \in I} (b_i + x_i) \leq \min_{j \in I} (a_j + y_j)$$

2. Selon un point de vue exécution, la non vacuité de l'intersection des intervalles absolus de validité temporelle n'est pas une condition suffisante pour supprimer tout risque de désynchronisation. En effet, bien que l'intersection des intervalles de validité temporelle des unités de synchronisation soit non vide, un problème de désynchronisation apparaît si la date de terminaison d'au moins une unité de synchronisation n'appartient pas à cette intersection. Ce cas de désynchronisation peut être exprimé formellement à partir des intervalles absolus de validité temporelle par le prédicat suivant:

$$(V_i \neq \emptyset) \wedge (\exists i \in I / \tau_i^{fin} \notin V_i)$$

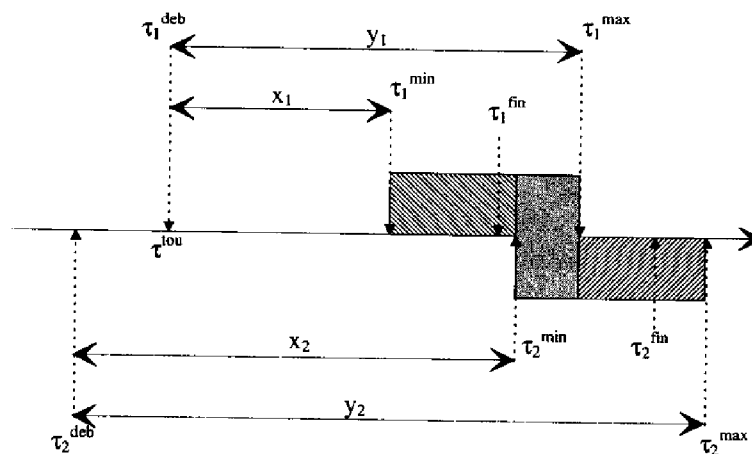


Figure 4.5 : Point de vue « exécution » d'un schéma de synchronisation inter-flux: cas où l'intersection des IAVT est non vide et les unités de synchronisation ne se terminent pas dans cette intersection.

Ainsi dans ce cas, tout instant de synchronisation appartenant à V_i n'est pas en mesure de garantir l'exécution complète de toutes les unités de synchronisation impliquées. De plus, tout instant de synchronisation appartenant à l'un quelconque des intervalles absolus de validité temporelle ne peut à la fois assurer l'exécution complète de chaque unité de synchronisation et garantir le respect des contraintes

temporelles de l'ensemble des unités de synchronisation impliquées dans ce schéma de synchronisation inter-flux. Ce cas de désynchronisation est illustré par le figure 4.5 où il apparaît que si l'instant de synchronisation inter-flux est choisi dans $[\tau_1^{\min}, \tau_1^{\max}] \cap [\tau_2^{\min}, \tau_2^{\max}]$ alors les contraintes de synchronisation de us_1 et de us_2 sont simultanément satisfaites mais l'exécution de us_2 est interrompue, ce qui peut poser problème si la présentation complète de l'unité d'information associée est indispensable.

3. Même le cas préjugé idéal où $(V_i \neq \emptyset) \wedge (\forall i \in I, \tau_i^{\text{fin}} \in V_i)$, est potentiellement générateur de problèmes de désynchronisation. En effet, bien que dans ce cas il existe un point de synchronisation dans V_i qui satisfasse à la fois les contraintes de synchronisation intra-flux et inter-flux, à savoir $\tau = \max_{i \in I}(\tau_i^{\text{fin}})$, il n'existe pas dans V_i de point de synchronisation susceptible de satisfaire, dans tous les cas, à la fois la stricte continuité² de l'ensemble des unités de synchronisation et leur exécution complète. Ainsi, par exemple, la figure 4.6 illustre un cas de synchronisation inter-flux pour lequel les deux unités de synchronisation considérées, us_1 et us_2 , se terminent dans l'intersection (non vide) de leurs intervalles de validité temporelle. Cependant quel que soit l'instant de synchronisation choisi dans cette intersection on ne peut assurer à la fois la terminaison correcte des deux unités de synchronisation ainsi que la stricte continuité entre la fin des deux unités de synchronisation et le début des unités de synchronisation suivantes.

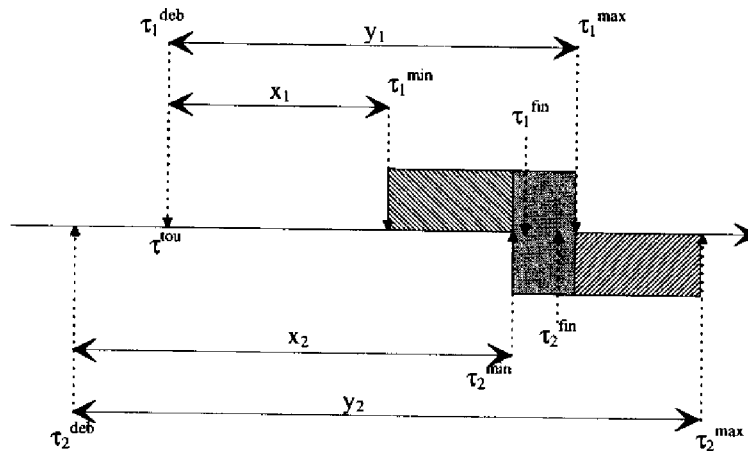


Figure 4.6 : Point de vue « exécution » d'un schéma de synchronisation inter-flux: cas où l'intersection des IAVT est non vide et les unités de synchronisation se terminent dans cette intersection

Les trois cas que nous venons d'aborder sont représentatifs de l'ensemble des problèmes de désynchronisation inter-flux susceptibles de survenir dans un environnement faiblement synchrone. Leur étude met en évidence que tout schéma de synchronisation inter-flux résulte d'un compromis entre les nécessités de préserver au mieux la

²La notion de stricte continuité met en jeu les unités de synchronisation qui doivent être exécutées à la suite du point de synchronisation considéré. En particulier, cette notion s'avère fondamentale pour la cohérence temporelle des schémas de synchronisation continus.

continuité des flux, d'assurer l'exécution complète des unités de synchronisation ainsi que de garantir le respect de leurs contraintes temporelles. Ainsi une sémantique de synchronisation en environnement faiblement synchrone doit offrir la possibilité de combiner précisément et de façon complète ces trois contraintes de synchronisation.

Nous venons de voir qu'en environnement faiblement synchrone tout schéma de synchronisation inter-flux est potentiellement générateur de problèmes de désynchronisation. Ainsi, une sémantique de synchronisation en environnement faiblement synchrone doit permettre de résoudre de façon explicite ces problèmes de désynchronisation. Cette sémantique de synchronisation doit permettre de définir de façon déterministe l'instant de synchronisation τ^{syn} associé à tout schéma de synchronisation inter-flux. Cet instant de synchronisation définit ainsi le comportement temporel de toutes les unités de synchronisation associées au point de synchronisation inter-flux considéré.

3.5 Vers une Sémantique de la Synchronisation en Environnement Faiblement Synchrone

Nous avons montré dans la section précédente que la mise en oeuvre de schémas de synchronisation inter-flux en environnement faiblement synchrone ne permet pas, dans le cas général, de garantir la sémantique temporelle, l'exécution complète, et la continuité de l'ensemble des flux mis en jeu. De plus, pour chaque schéma de synchronisation inter-flux, plusieurs points de synchronisation potentiels apparaissent comme candidats et induisent par la même autant de sémantiques de synchronisation. Chacun de ces points de synchronisation permet, en fonction d'une certaine sémantique applicative ou de la spécificité de certains flux, de préserver la sémantique temporelle, la terminaison complète, ou la continuité de certaines unités de synchronisation. Ainsi, afin de spécifier de façon non ambiguë des scénarios de synchronisation, il est nécessaire d'introduire une sémantique précise de la synchronisation en environnement faiblement synchrone qui explicite l'ensemble des instants susceptibles d'être associés de façon cohérente à un point de synchronisation inter-flux.

L'impossibilité de pouvoir garantir dans tous les cas les contraintes temporelles, la terminaison complète et la continuité de l'ensemble des unités de synchronisation impliquées dans un schéma de synchronisation inter-flux induit des stratégies de synchronisation visant à préserver une unité de synchronisation pré-déterminée.

Ainsi trois stratégies de synchronisation fondamentales et orthogonales peuvent être dégagées:

- Une stratégie de synchronisation conduite par *l'unité de synchronisation la plus en avance*. Par unité de synchronisation la plus en avance on entend, selon un point de vue exécution, celle se terminant au plus tôt, et selon un point de vue modélisation celle atteignant au plus tôt la borne supérieure de son intervalle de validité temporelle. Cette unité de synchronisation est définie dynamiquement en fonction des dérivées temporelles qui sont apparues dans les flux auxquels appartiennent les unités de synchronisation considérées. Une telle stratégie de synchronisation garantit la cohérence temporelle d'au moins une unité de synchronisation parmi l'ensemble des unités de synchronisation devant être synchronisées. De plus, cette stratégie induit l'accélération ou à l'extrême la terminaison « forcée » des unités de synchronisation « en retard ». Ainsi, ce type de synchronisation n'est pas en mesure de garantir le respect des contraintes

temporelles des unités de synchronisation en retard. Une telle stratégie de synchronisation est à privilégier si, avant toute chose, on ne veut induire aucun dépassement de date de terminaison au plus tard pour les unités de synchronisation du schéma de synchronisation considéré et celles leur succédant. Il convient de noter que, selon un point de vue exécution, pour un schéma de synchronisation donné il existe une infinité de sémantiques de synchronisation susceptibles de satisfaire une telle stratégie de synchronisation. En effet, considérons à titre d'exemple le schéma de synchronisation illustré par la figure 4.5. Tous les instants de synchronisation appartenant à l'intervalle $[\tau_1^{fin}, \tau_1^{max}]$ satisfont les contraintes temporelles de l'unité de synchronisation us_1 . Dans cet intervalle seul l'instant τ_1^{fin} permet de satisfaire simultanément les contraintes temporelles et la continuité du flux associé à us_1 . A l'opposé l'instant τ_1^{max} ne permet pas de satisfaire la continuité de us_1 , mais par contre ce point de synchronisation permet de prolonger l'exécution de us_2 tout en satisfaisant les contraintes temporelles de us_1 .

- Une stratégie de synchronisation qui garantit le respect des contraintes temporelles de *l'unité de synchronisation la plus en retard*. Cette unité de synchronisation correspond, selon un point de vue exécution, à celle se terminant au plus tard, et selon un point de vue modélisation à celle atteignant au plus tard la borne supérieure de son intervalle de validité temporelle. Tout comme pour la stratégie de synchronisation précédente l'unité de synchronisation « dirigeant » la synchronisation est définie dynamiquement en fonction des dérives inter-flux. De plus cette stratégie de synchronisation peut induire le ralentissement, et à l'extrême peut induire l'incohérence temporelle, des unités de synchronisation « en avance ». Une telle stratégie de synchronisation doit être privilégiée lorsqu'on souhaite assurer, en tout premier lieu, l'exécution complète de toutes les unités de synchronisation impliquées dans un schéma de synchronisation inter-flux. Selon un point de vue exécution, pour un schéma de synchronisation donné il existe une infinité de sémantiques de synchronisation susceptibles de satisfaire une telle stratégie de synchronisation. En effet, considérons à nouveau le schéma de synchronisation illustré par la figure 4.5. Tous les instants de synchronisation appartenant à l'intervalle $[\tau_2^{min}, \tau_2^{fin}]$ satisfont les contraintes temporelles de l'unité de synchronisation la plus en retard, à savoir us_2 . En particulier, l'instant τ_2^{fin} permet l'exécution complète de us_1 et de us_2 . A l'opposé l'instant τ_2^{min} ne permet pas d'assurer la terminaison complète de us_2 , mais par contre ce point de synchronisation permet de minimiser la dérive inter-flux tout en assurant le respect des contraintes temporelles du flux le plus en retard.
- Une stratégie de synchronisation qui assure le respect des contraintes temporelles *d'un flux, dit flux maître, pré-déterminé statiquement parmi l'ensemble des flux*. Cette stratégie induit « l'accélération » des unités de synchronisation en retard par rapport à l'unité de synchronisation « maîtresse »; de même elle implique le « ralentissement » des unités de synchronisation en avance. Une telle stratégie de synchronisation est à privilégier lorsque le respect des contraintes de synchronisation de l'unité de synchronisation « maîtresse » est fondamental pour la sémantique applicative. Selon un point de vue exécution cette stratégie de synchronisation est susceptible d'induire une infinité de points de synchronisation permettant de privilégier en fonction des besoins applicatifs ou bien la continuité de l'unité de synchronisation maîtresse, ou bien la dérive inter-flux par rapport aux unités de synchronisation en avance, ou bien la terminaison complète des unités de synchronisation en retard.

A l'identique de la solution proposée dans [Ste90], selon un point de vue « exécution », un délai d'attente conditionnel peut être associé à chaque unité de synchronisation impliquée dans un point de synchronisation inter-flux. Ce délai d'attente conditionnel spécifie, dans le cas où un « rendez-vous » ne peut pas être immédiatement satisfait, combien de temps une unité de synchronisation est prête à attendre ainsi que l'action à réaliser pendant cette attente.

Ainsi, il apparaît qu'une sémantique formelle de la synchronisation en environnement faiblement synchrone doit être définie par:

- selon un point de vue exécution, des fonctions ou opérateurs permettant d'obtenir de façon déterministe un instant de synchronisation τ^{syn} unique à partir des instants de synchronisation fondamentaux (τ_i^{min} , τ_i^{max} , τ_i^{fin} , τ^{tra}) des unités de synchronisation impliquées dans un schéma de synchronisation intra ou inter-flux.
- selon un point de vue modélisation, des fonctions ou opérateurs permettant d'obtenir à partir des instants de synchronisation fondamentaux (τ_i^{min} , τ_i^{max} , τ^{tra}) tous les instants de synchronisation τ^{syn} potentiellement réalisables. Tout comme la notion d'intervalle de validité temporelle recouvre tous les instants potentiels de terminaison d'une unité de synchronisation, ce point de vue introduit la notion d'intervalle de synchronisation (ou intervalle de tir dans la terminologie des réseaux de Petri) qui explicite sous une forme abstraite tous les instants potentiels de synchronisation associés à un point de synchronisation.

Etant donné le nombre potentiellement infini de sémantiques de synchronisation répondant, selon un point de vue exécution, à la classification des types fondamentaux de synchronisation que nous avons introduite, nous adopterons dorénavant, afin de définir une sémantique de synchronisation la plus abstraite possible, un point de vue modélisation.

La définition d'une sémantique de synchronisation selon un point de vue modélisation, consiste alors à définir les opérateurs permettant d'obtenir, à partir des intervalles de validité temporelle des unités de synchronisation considérées, les intervalles de synchronisation définissant les trois sémantiques de synchronisation fondamentales pouvant être associées à un point de synchronisation inter-flux. Ces trois sémantiques fondamentales de synchronisation sont spécifiées par trois opérateurs sur les intervalles absolus de validité temporelles notés « ou-fort », « et-faible », et « maître », et formellement définis comme il suit:

$$\begin{aligned} \text{ou-fort}([\tau_i^{\text{min}}, \tau_i^{\text{max}}]_{i \in I}) &= [\min_i(\tau_i^{\text{min}})_{i \in I}, \min_i(\tau_i^{\text{max}})_{i \in I}] \\ \text{et-faible}([\tau_i^{\text{min}}, \tau_i^{\text{max}}]_{i \in I}) &= [\max_i(\tau_i^{\text{min}})_{i \in I}, \max_i(\tau_i^{\text{max}})_{i \in I}] \\ \text{maître}(m)([\tau_i^{\text{min}}, \tau_i^{\text{max}}]_{i \in I}) &= [\tau_m^{\text{min}}, \tau_m^{\text{max}}] \text{ avec } m \in I \end{aligned}$$

Il apparaît de façon immédiate que les opérateurs « ou-fort », « et-faible » et « maître » définissent respectivement les stratégies de synchronisation conduites par l'unité de synchronisation la plus en avance, la plus en retard, et par l'unité de synchronisation maîtresse d'indice m .

		Borne supérieure de l'intervalle de synchronisation déterminée par l'unité de synchronisation :		
		<i>La plus en avance</i>	<i>La plus en retard</i>	<i>maître</i>
Borne inférieure de l'intervalle de synchronisation déterminée par l'unité de synchronisation :	<i>La plus en avance</i>	ou-fort	ou	ou-maître
	<i>La plus en retard</i>	et	et-faible	et-maître
	<i>maître</i>	maître-fort	maître-faible	maître

Tableau 4-2 : Définition des 9 sémantiques de synchronisation en environnement faiblement synchrone.

Ces trois stratégies peuvent être combinées pour définir un jeu complet de 9 opérateurs de synchronisation. En effet, le calcul de tous les intervalles temporellement cohérents obtenus en appliquant les opérateurs min/max/maître sur les bornes des intervalles absolus de validité temporelle permet de définir 9 intervalles de synchronisation utilisant sélectivement une stratégie « et-faible », « ou-fort », ou « maître » pour la définition de chacune des bornes de l'intervalle de synchronisation considéré (tableau 4.2) . Nous considérons que les 9 opérateurs de synchronisation ainsi définis définissent de façon complète les sémantiques de synchronisation de base envisageables dans un environnement faiblement synchrone.

Ces 9 sémantiques de synchronisation sont définies formellement par les opérateurs suivants:

$$\begin{aligned}
 \text{ou - fort}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\min_i(\tau_i^{\min})_{i \in I}, \min_i(\tau_i^{\max})_{i \in I} \right] \\
 \text{ou}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\min_i(\tau_i^{\min})_{i \in I}, \max_i(\tau_i^{\max})_{i \in I} \right] \\
 \text{et - faible}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\max_i(\tau_i^{\min})_{i \in I}, \max_i(\tau_i^{\max})_{i \in I} \right] \\
 \text{et}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\max_i(\tau_i^{\min})_{i \in I}, \max\left(\max_i(\tau_i^{\min})_{i \in I}, \min_i(\tau_i^{\max})_{i \in I}\right) \right] \\
 \text{maître}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= [\tau_m^{\min}, \tau_m^{\max}] \text{ avec } a_m \text{ l' arc maître} \\
 \text{ou - maître}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\min_i(\tau_i^{\min})_{i \in I}, \tau_m^{\max} \right] \\
 \text{et - maître}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\max_i(\tau_i^{\min})_{i \in I}, \max\left(\max_i(\tau_i^{\min})_{i \in I}, \tau_m^{\max}\right) \right] \\
 \text{maître - faible}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\tau_m^{\min}, \max_i(\tau_i^{\max})_{i \in I} \right] \\
 \text{maître - fort}([\tau_i^{\min}, \tau_i^{\max}])_{i \in I} &= \left[\tau_m^{\min}, \max\left(\min_i(\tau_i^{\max})_{i \in I}, \tau_m^{\min}\right) \right]
 \end{aligned}$$

Quelles que soient les valeurs des intervalles absolus de validité temporelle des unités de synchronisation que l'on souhaite synchroniser (en particulier, même si leur intersection est vide), ces 9 opérateurs permettent d'adopter un type de synchronisation inter-flux prenant en compte de façon précise la sémantique applicative ou les spécificités des flux. Ces 9 opérateurs de synchronisation sont illustrés par la figure 4.7 qui représente un

schéma générique de synchronisation inter-flux mettant en jeu 3 unités de synchronisation modélisées par (P_4, t_5) , (P_5, t_5) et (P_6, t_5) . Le caractère générique de ce schéma de synchronisation est assuré par les trois arcs (P_1, t_2) , (P_2, t_3) et (P_4, t_4) qui modélisent les retards préalables aux trois unités de synchronisation considérées.

La figure 4.7-b représente, en fonction des intervalles absolus de validité temporelle des trois unités de synchronisation, l'intervalle de synchronisation défini par chacun des 9 opérateurs de synchronisation. On notera que ces 9 opérateurs de synchronisation donnent des intervalles de synchronisation identiques lorsque le point de synchronisation considéré se réduit à une seule unité de synchronisation. Cet intervalle se réduisant alors à celui défini par la sémantique de tir des réseaux de Petri Temporels.

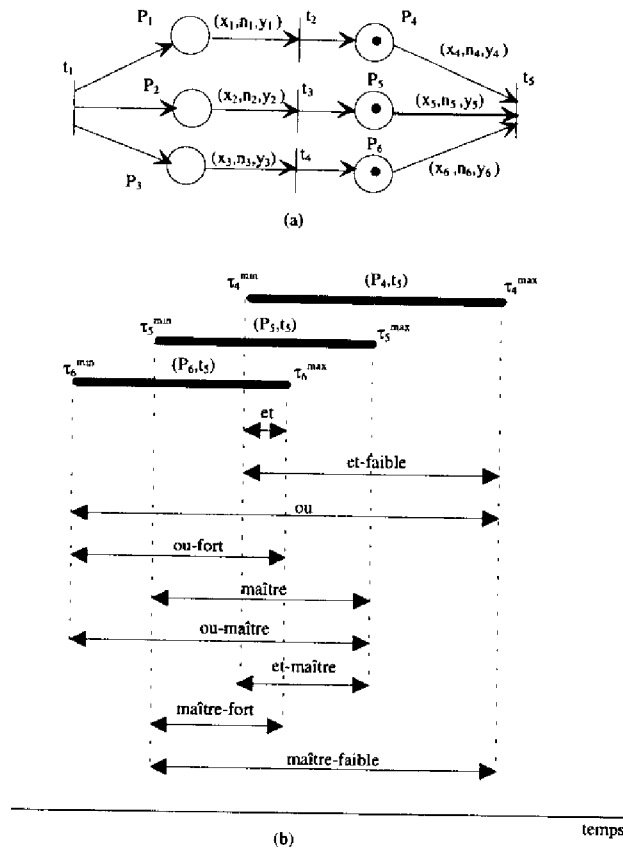


Figure 4.7: Les intervalles de synchronisation définis par les 9 opérateurs de synchronisation. (a) Un schéma de synchronisation inter-flux mettant en jeu trois unités de synchronisation, respectivement (P_4, t_5) , (P_5, t_5) et (P_6, t_5) . (b) Les intervalles de synchronisation définis en appliquant chacun des 9 opérateurs fondamentaux de synchronisation sur les intervalles absolus de validité temporelle des trois unités de synchronisation considérées. L'unité de synchronisation (P_5, t_5) est considérée comme « maîtresse » pour les opérateurs « maître », « ou-maître », « et-maître », « maître-fort », « maître-faible ».

Cependant, les 9 opérateurs de synchronisation ainsi définis sont susceptibles d'induire le tir de transitions partiellement sensibilisées. Il en est ainsi, par exemple, pour le scénario de synchronisation spécifié par la figure

4.8, si nous associons au point de synchronisation inter-flux modélisé par la transition t_5 une sémantique de synchronisation de type « maître » avec l'arc (P_5, t_5) comme arc « maître ». Dans ce cas, si les unités de synchronisation (P_1, t_2) et (P_3, t_4) finissent à leur date de terminaison au plus tard, *i.e.* 7, et si (P_2, t_3) se termine à sa date au plus tôt, *i.e.* 0, alors les intervalles absolus de validité temporelle des trois unités de synchronisation (P_4, t_5) , (P_5, t_5) et (P_6, t_6) , impliquées dans le point de synchronisation inter-flux considéré, sont respectivement $(7, 8, 9)$, $(4, 5, 6)$ et $(7, 8, 9)$. Si on applique à ces trois intervalles l'opérateur « maître » en considérant l'arc (P_5, t_5) comme arc maître, on obtient l'intervalle de synchronisation $(4, 5, 6)$. Ainsi quel que soit l'instant de synchronisation considéré dans cet intervalle, la transition t_5 n'est pas sensibilisée. Le strict respect de la sémantique temporelle de l'arc « maître » (P_5, t_5) nécessiterait de tirer une transition partiellement sensibilisée.

Il est donc nécessaire d'introduire une sémantique de synchronisation résolvant ces cas de désynchronisation. Trois catégories de solutions peuvent être proposées pour résoudre ce problème:

- Offrir une méthode de vérification permettant de mettre a priori en évidence les risques de désynchronisation dans un scénario de synchronisation. [Wal83] suggère une telle méthode qui débouche malheureusement dans le cas de réseaux de Petri de complexité arbitraire sur des algorithmes de complexité non-polynomiale. Nous verrons cependant par la suite que l'on peut proposer des algorithmes de complexité polynomiale pour des réseaux utilisant exclusivement les structures de synchronisation de Allen (voir chapitre 5).
- Introduire une sémantique de tir permettant le tir de transitions partiellement sensibilisées. Une telle solution pose le problème de l'identification et de la gestion des jetons « en retard ». En particulier, l'identification des jetons « en retard » nécessite, tout comme dans la solution précédente, la détermination de l'événement le plus récent susceptible d'induire la sensibilisation de la transition considérée. Ainsi cette solution offre le double désavantage d'être à la fois éloignée de la sémantique de synchronisation des réseaux de Petri de base, et de déboucher dans le cas général sur des algorithmes de complexité non-polynomiale.
- Relâcher les contraintes de synchronisation en introduisant des intervalles de synchronisation débutant au plus tôt lorsque la transition considérée est sensibilisée. Une telle solution introduit une sémantique de synchronisation de type « best-effort » qui s'efforce de respecter au mieux les sémantiques de synchronisation précédemment introduites. Nous retiendrons cette solution car elle apporte une réponse simple, effectivement réalisable dans tous les cas, et conforme à la synchronisation de type « best-effort » induite par les systèmes faiblement synchrones. En effet plusieurs arguments militent en faveur d'une telle approche; d'une part il ne semble pas cohérent de rejeter des schémas de synchronisation, à l'identique de la figure 4.8, qui sont potentiellement exposés à des dérives temporelles intempestives mais qui cependant sont, dans le cas nominal, parfaitement synchrones; d'autre part des contraintes sur les ressources disponibles ainsi que la nécessité d'avoir des spécifications portables ne permettent pas toujours de modifier la granularité de synchronisation intra et inter-flux afin de réduire les risques de désynchronisation. Aussi est-il nécessaire de proposer une sémantique de synchronisation de type « best effort » induisant un comportement robuste et cohérent face à tout problème de désynchronisation.

Cependant il est important de garder à l'esprit que, dans le cadre d'applications « temps-réel dur », des techniques de vérification peuvent s'avérer indispensables afin de mettre en évidence tout risque de

désynchronisation. Ces techniques de vérification doivent permettre de mettre en oeuvre exclusivement des schémas de synchronisation satisfaisant dans tous les cas les sémantiques de synchronisation « fortes » précédemment introduites.

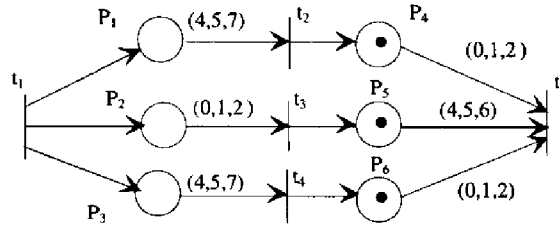


Figure 4.8: Un scénario de synchronisation inter-flux ayant un caractère générique.

Ainsi cette approche de type « best-effort » nécessite la redéfinition des 9 opérateurs de synchronisation sous la forme suivante:

$$\begin{aligned}
 \text{ou - fort}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \min(\tau_i^{\min})_{i \in I}), \max(\tau^{\text{tou}}, \min(\tau_i^{\max})_{i \in I}) \right] \\
 \text{ou}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \min(\tau_i^{\min})_{i \in I}), \max(\tau_i^{\max})_{i \in I} \right] \\
 \text{et - faible}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau_i^{\min})_{i \in I}, \max(\tau_i^{\max})_{i \in I} \right] \\
 \text{et}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau_i^{\min})_{i \in I}, \max(\max(\tau_i^{\min})_{i \in I}, \min(\tau_i^{\max})_{i \in I}) \right] \\
 \text{maître}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \tau_m^{\min}), \max(\tau^{\text{tou}}, \tau_m^{\max}) \right] \text{ avec } a_m \text{ l'arc maître} \\
 \text{ou - maître}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \min(\tau_i^{\min})_{i \in I}), \max(\tau^{\text{tou}}, \tau_m^{\max}) \right] \\
 \text{et - maître}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau_i^{\min})_{i \in I}, \max(\max(\tau_i^{\min})_{i \in I}, \tau_m^{\max}) \right] \\
 \text{maître - faible}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \tau_m^{\min}), \max(\tau_i^{\max})_{i \in I} \right] \\
 \text{maître - fort}([\tau_i^{\min}, \tau_i^{\max}]_{i \in I}) &= \left[\max(\tau^{\text{tou}}, \tau_m^{\min}), \max(\tau^{\text{tou}}, \min(\tau_i^{\max})_{i \in I}, \tau_m^{\min}) \right]
 \end{aligned}$$

Ces 9 opérateurs de synchronisation sont obtenus en rajoutant, si nécessaire, des contraintes sur les bornes des intervalles de synchronisation définis précédemment de façon à obtenir des intervalles temporels débutant au plus tôt à la date de sensibilisation de la transition, soit l'instant τ^{tou} . Ainsi chacun de ces 9 opérateurs fournit une sémantique de synchronisation conforme à la sémantique de tir des réseaux de Petri et couvrant, à la différence des réseaux de Petri à flux temporels, tous les cas de désynchronisation.

4. Définition Formelle des RdPFT

Un réseau de Petri étendu pour la spécification des contraintes de synchronisation en environnement faiblement synchrone doit donc offrir les caractéristiques suivantes:

- permettre de spécifier les contraintes temporelles des unités de synchronisation sous la forme d'arcs associés à un intervalle de validité temporelle. Cette approche permet une spécification complète des contraintes de synchronisation intra-flux. De plus, l'utilisation d'arcs temporels induit un pouvoir de modélisation permettant une spécification aisée des contraintes de synchronisation inter-flux à partir d'un paradigme de composition de flux inspiré de celui de la « régie de production ».

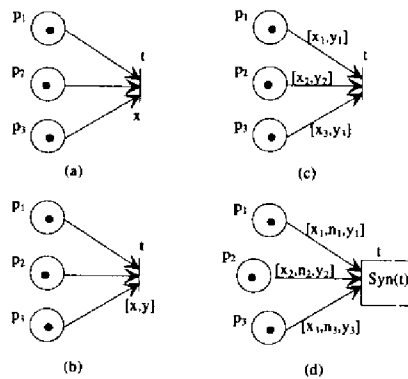


Figure 4.9: Les Réseaux de Petri à Flux Temporels par rapport aux réseaux de Petri à extensions temporelles préexistants. (a) réseau de Petri temporisé. (b) réseau de Petri temporel. (c) réseaux de Petri à arcs temporels. (d) réseau de Petri à flux temporels.

- permettre d'associer à chacune des transitions modélisant un point de synchronisation inter-flux l'une quelconque des 9 sémantiques de synchronisation précédemment introduites. Une telle approche confère un pouvoir d'expression permettant de prendre en compte de façon complète et précise les problèmes de désynchronisation dans un système faiblement synchrone.

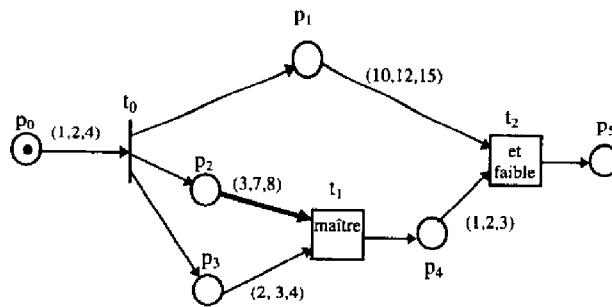


Figure 4.10: Exemple de RdPFT

Ces constatations nous conduisent à introduire un nouveau modèle, appelé Réseau de Petri à Flux Temporels (RdPFT) (figure 4.9-d) [DS93b][SD94], ayant un pouvoir de modélisation et/ou d'expression supérieur aux extensions temporelles préexistantes.

Les notations définies en Annexe seront utilisées par la suite.

Définition 4.1: Un Réseau de Petri à Flux Temporels est un n-uplet $(P, T, \text{Pré}, \text{Post}, M_0, \text{IVT}, \text{Syn}, \text{AM})$, avec:

- (P, T, F, B, M_0) définissant un réseau de Petri,
- IVT est la fonction qui permet d'associer un intervalle de validité temporelle à chaque arc sortant d'une place. Cette fonction est définie de la façon suivante:

$$\text{IVT}: A^+ \rightarrow (Q^+ \cup +\infty) \times (Q^+ \cup +\infty) \times (Q^+ \cup +\infty)$$

$$\forall a_i \in A^+, \text{IVT}(a_i) = (x_i, n_i, y_i) / x_i \leq n_i \leq y_i$$

La représentation graphique d'un RdPFT étiquette chaque arc sortant d'une place à l'aide de son intervalle de validité temporelle (figure 4.9-d).

- Syn est une fonction de typage de transition qui associe une sémantique de synchronisation à chaque transition. Cette fonction est définie de la façon suivante:

$$\text{Syn}: T \rightarrow \{\text{et}, \text{et-faible}, \text{ou}, \text{ou-fort}, \text{maître}, \text{maître-faible}, \text{maître-fort}, \text{et-maître}, \text{ou-maître}\}$$

Notons que pour les transitions ayant un préfixe constitué d'une seule place toutes ces sémantiques de synchronisation sont égales. La représentation graphique d'un RdPFT associe à chaque transition l'identificateur de la sémantique de synchronisation qui lui est associée. Cet identificateur est omis pour les transitions ayant un préfixe réduit à une seule place (figure 4.10).

- AM est une fonction qui associe à toute transition de type « maître », « maître-fort », « maître-faible », « et-maître », « ou-maître », un arc « maître » appartenant à l'ensemble des arcs entrant de la transition considérée. Cette fonction est définie de la façon suivante:

$$\text{AM}: T \rightarrow A^+$$

$$\forall t_k \in T / \text{Syn}(t_k) \in \{\text{maître}, \text{maître-fort}, \text{maître-faible}, \text{et-maître}, \text{ou-maître}\}$$

$$\exists! a \in A_k^+ / \text{AM}(t_k) = a$$

Les arcs « maîtres » sont représentés graphiquement par des arcs en trait gras (figure 4.10).

De plus on supposera qu'une horloge notée H est associée à tout RdPFT.

5. Sémantique Formelle des RdPFT

Nous allons introduire dans cette section la notion d'état d'un RdPFT. L'état d'un RdPFT permet de décrire à tout instant l'état temporel de l'ensemble des unités de synchronisation spécifiées par le RdPFT considéré. De plus, cette notion associée à des règles d'évolution entre états permet de décrire l'évolution dynamique du

système modélisé. Ainsi la formalisation du concept d'état d'un RdPFT couplée à des règles formelles d'évolution entre états définit la sémantique formelle des RdPFT.

5.1 Etat d'un RdPFT

La notion d'état d'un RdPFT peut être introduite en s'inspirant de la notion d'état d'un réseau de Pétri temporel donnée par [BD91]. Afin de ne pas introduire des notations trop complexes, nous considérerons par la suite des RdPFT pour lesquels les arcs ne peuvent être multi-sensibilisés³, autrement dit on ne considérera que des RdPFT « saufs ».

L'état d'un RdPFT est donné à tout instant par un couple $S=(M,I)$ tel que:

- M représente le marquage courant du RdPFT,
- I est la liste des intervalles dynamiques des arcs sensibilisés par le marquage courant. La notion d'intervalle dynamique sera explicitée par les règles d'évolution entre états données dans la section suivante. Le nombre d'éléments de cette liste est égal au nombre d'arcs sensibilisés par le marquage courant. Ainsi, la cardinalité de cet ensemble évolue à travers l'espace des états. De plus, cette liste est ordonnée selon l'ordre lexicographique des identificateurs d'arcs.

Considérons à titre d'exemple le RdPFT représenté dans la figure 4.10. L'état initial de ce RdPFT est donné par le couple $S_0=(M_0,I_0)$ tel que:

- $M_0=(p_0(1))$ avec la notation $p_0(1)$ indiquant que la place p_0 est marquée par un jeton ,
- $I_0=((1,2,4))$

La seule transition sensibilisée dans l'état S_0 est t_0 . Cette transition est tirable dans l'intervalle relatif (*i.e.* relatif à la date initiale associée à l'occurrence de l'état S_0) de validité temporelle (1,2,4). Si la transition t_0 est tirée à l'instant relatif $\theta=4$, alors ce RdPFT passe dans l'état $S_1=(M_1,I_1)$ tel que:

- $M_1=(p_1(1),p_2(1),p_3(1))$
- $I_1=((10,12,15),(3,7,8),(2,3,4))$

5.2 Règles d'Evolution entre Etats

Supposons qu'une transition t soit tirable dans l'état $S=(M,I)$ à l'instant relatif θ par rapport à la date d'occurrence de cet état. Alors l'état $S'=(M',I')$ atteint à partir de l'état S en tirant la transition t à l'instant θ est calculé de la façon suivante:

- Le nouveau marquage est calculé d'une façon identique aux réseaux de Petri de base, à savoir:

$$(\forall p \in P) M'(p) = M(p) - \text{Pré}(p,t) + \text{Post}(p,t)$$

³Par définition, un arc $a=(p,t)$ est sensibilisé si et seulement $M(p) \geq B(p,t)$.

- L'ensemble I' est calculé selon les trois étapes suivantes:
 1. Retirer de l'ensemble I tous les intervalles dynamiques (*i.e.* tous les triplets) correspondant aux arcs qui étaient sensibilisés dans l'état S et qui ne le sont plus suite au tir de la transition t .
 2. Réduire d'une valeur θ vers l'origine des temps les bornes (*i.e.* bornes minimum, maximum et durée nominale) de tous les intervalles dynamiques de validité temporelle restant dans l'ensemble I suite à l'étape précédente. Borner par zéro les valeurs des nouvelles bornes minimum et maximum ainsi obtenues. Ainsi l'intervalle dynamique (x,n,y) est remplacé dans I par $(\max(0, x-\theta), n-\theta, \max(0, y-\theta))$.
 3. Introduire dans l'ensemble I les intervalles de validité temporelle des arcs nouvellement sensibilisés ou resensibilisés suite au tir de t .

Les intervalles contenus dans l'ensemble I associé à l'état S ont donc des bornes qui évoluent dynamiquement en fonction de la date relative de tir de la transition t . Ainsi, ces intervalles sont appelés intervalles dynamiques de validité temporelle. L'intervalle dynamique de validité temporelle (x,n,y) d'un arc, noté a , sensibilisé dans l'état S , spécifie respectivement l'instant relatif⁴ de terminaison au plus tôt, le retard (si $n \leq 0$) ou l'avance (si $n \geq 0$) par rapport à la durée nominale, et l'instant relatif de terminaison au plus tard de cette unité de synchronisation.

5.3 Règles de Tir des RdPFT

Après avoir introduit la notion d'état d'un RdPFT ainsi que les règles d'évolution entre états mises en oeuvre suite au tir d'une transition, nous allons introduire dans cette section les règles de tir formelles d'une transition dans un état donné d'un RdPFT. Ces règles de tir induisent 9 sémantiques de synchronisation en fonction du type de synchronisation associé à la transition considérée par l'intermédiaire de la fonction de synchronisation « Syn ».

5.3.1 Règles de Tir pour une Transition de Type « et-faible »

La synchronisation de type « et-faible » a pour objectif, dans un schéma de synchronisation inter-flux, d'assurer l'exécution complète de l'ensemble des unités de synchronisation considérées. Ainsi l'instant de synchronisation associé à un point de synchronisation de type « et-faible » est déterminé à partir de la date de terminaison de l'unité de synchronisation la plus en retard. Cette unité de synchronisation est définie dynamiquement en fonction des dérivées temporelles cumulées dans les unités de synchronisation « amont » conduisant à la sensibilisation des arcs considérés. Une telle stratégie de synchronisation ne permet pas évidemment de garantir la correction temporelle des unités de synchronisation « en avance ». Ainsi, celles-ci sont « ralenties » par l'unité de synchronisation la plus en retard.

Nous avons vu que cette sémantique de synchronisation peut être définie par un opérateur sur les intervalles absolus de validité temporelle des arcs considérés. La progression du temps, telle qu'elle est définie dans les

⁴ Cette durée est relative à la date d'occurrence de l'état considéré.

règles d'évolution entre états, nous permet d'exprimer d'une manière équivalente l'intervalle de tir d'une transition à partir des intervalles dynamiques de validité temporelle des arcs.

Définition 4.2: Une transition t_k telle que $\text{Syn}(t_k)=\text{et-faible}$ et $\forall a_i \in A_k^+, \text{IDVT}(a_i) = (\alpha_i, d_i, \beta_i)$, est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée, c'est à dire que la propriété suivante est vérifiée:

$$(\forall p \in P)(M(p) \geq \text{Pré}(p, t_k))$$

- l'instant relatif θ satisfait les inégalités suivantes:

$$\text{MIN}_k = \max_i(\alpha_i) \leq \theta \leq \max_i(\beta_i) = \text{MAX}_k$$

Avec l'indice i parcourant l'ensemble I_k .

L'utilisation de cette sémantique de synchronisation doit être favorisée dans des schémas de synchronisation inter-flux où l'on veut assurer l'exécution complète de l'ensemble des unités de synchronisation.

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « et-faible » peut être défini par l'exécutif de la façon suivante:

$$\tau^{\text{syn}} = \min \left\{ \tau \in H / \left(\forall i \in I_k, \left((\tau \geq \tau_i^{\text{min}}) \wedge (\tau \geq \tau_i^{\text{fin}}) \right) \vee (\tau \geq \tau_i^{\text{max}}) \right) \right\}$$

Considérons à titre d'exemple le réseau de Petri à arcs temporel de la figure 4.8. Ce scénario de synchronisation est susceptible d'induire entre les unités de synchronisation (P_4, t_5) , (P_5, t_5) et (P_6, t_5) une désynchronisation telle que l'intersection de leurs intervalles de validité temporelle soit vide. En particulier ceci est vrai si nous supposons que les dates de terminaison⁵ des unités de synchronisation (P_1, t_2) , (P_3, t_3) et (P_4, t_4) sont respectivement 7, 0 et 7. La sémantique de synchronisation des réseaux de Petri à arcs temporels s'avère impuissante à résoudre ce problème de désynchronisation. Par contre, si nous considérons ce réseau comme un RdPFT, et si nous associons à la transition t_5 une sémantique de synchronisation de type « et-faible » alors, pour les dates de terminaison données, l'intervalle de tir (relativement à l'état atteint lorsque t_5 devient sensibilisée) associé à la transition t_5 est $[\text{MIN}_5, \text{MAX}_5]=[0,2]$. De même en raisonnant à l'aide des intervalles absolus de validité temporelle (en supposant que le scénario débute au temps absolu $\tau=0$) et en utilisant l'opérateur de synchronisation « et-faible » défini dans la section 2.5 de ce chapitre on obtient un intervalle de synchronisation égal à $[\max(7,4,7), \max(9,6,9)]=[7,9]$. Dans ce cas de figure l'unité de synchronisation (P_5, t_5) , en « avance » par rapport aux deux autres, sera retardée au point d'être temporellement incorrecte. La dérive temporelle imposée à l'unité de synchronisation (P_3, t_5) retardée est comprise entre $7-6=1$ et $9-4=5$ unités de temps.

⁵ Pour ces dates de terminaison nous obtenons suite aux tirs des transitions t_2, t_3, t_4 du RdPFT de la Figure 4.8 l'état $S_3=(M_3, I_3)$ tel que $M_3=(p_4(1), p_5(1), p_6(1))$ et $I_3=((0,1,2), (0,-2,0), (0,1,2))$. De même dans ce cas les intervalles absolus de validité temporelle des arcs (p_4, t_5) , (p_5, t_5) et (p_6, t_5) sont respectivement $(7,8,9)$, $(4,5,6)$ et $(7,8,9)$.

Si nous adoptons un point de vue exécution et considérons que (P_4, t_5) , (P_5, t_5) et (P_6, t_5) se terminent respectivement aux dates absolues 9, 6 et 9, alors, conformément à l'implémentation de la sémantique « et-faible » donnée ci-dessus, l'instant de synchronisation est égal à $\tau^{syn}=9$.

5.3.2 Règles de Tir pour une Transition de Type « et »

La sémantique de synchronisation de type « et » a pour objectif de satisfaire « pour le mieux » les contraintes temporelles de l'ensemble des unités de synchronisation devant être synchronisées. Les contraintes temporelles de l'ensemble des unités de synchronisation impliquées dans un schéma de synchronisation inter-flux ne peuvent être satisfaites que si et seulement si l'intersection de leurs intervalles statiques de validité temporelle est non vide. Ainsi dans le cas où cette intersection est non-vide la sémantique de synchronisation de type « et » est identique à celle des réseaux de Petri à arcs temporels. Dans ce cas l'intervalle de tir est défini comme étant l'intersection des intervalles absolus de validité temporelle. Cependant, la sémantique de synchronisation de type « et » proposée ici lève les limitations des règles de tir des réseaux de Petri à arcs temporels, en résolvant le cas où l'intersection des intervalles absolus de validité temporelle est vide. Dans ce cas, cette sémantique de synchronisation essaie de satisfaire « au mieux » l'ensemble des contraintes temporelles en réduisant l'intervalle de synchronisation à un instant de synchronisation égal au maximum des bornes inférieures des intervalles absolus de validité temporelle. Ainsi, dans tous les cas, toutes les unités de synchronisation se seront exécutées au moins jusqu'à concurrence de leur durée minimale d'exécution.

Ce type de synchronisation peut induire le « ralentissement » des unités de synchronisation « en avance » ainsi que « l'accélération » ou l'exécution incomplète des unités de synchronisation « en retard ».

Définition 4.3: Une transition t_k telle que $Syn(t_k)=et$, et $\forall a_i \in A_k^+, IDVT(a_i) = (\alpha_i, d_i, \beta_i)$, est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$MIN_k = \max_i(\alpha_i) \leq \theta \leq \max(\min_i(\beta_i), \max_i(\alpha_i)) = MAX_k$$

Avec l'indice i parcourant l'ensemble I_k .

On peut noter que le cas où l'intersection des intervalles absolus de validité temporelle est vide se traduit en terme d'intervalles dynamiques par une intersection se réduisant au maximum des bornes inférieures.

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « et » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min \left\{ \begin{array}{l} \tau \in H / \left(\forall i \in I_k, (\tau \geq \tau_i^{min}) \wedge (\tau \geq \tau_i^{fn}) \right) \\ \vee \\ \left((\exists j \in I_k / \tau \geq \tau_j^{max}) \wedge (\forall i \in I_k, \tau \geq \tau_i^{min}) \right) \end{array} \right\}$$

Considérons à titre d'exemple le scénario de synchronisation de la figure 4.8, et associons à la transition t_5 une sémantique de synchronisation de type « et ». Si nous supposons que les unités de synchronisation (P_1, t_2) , (P_3, t_3) et (P_4, t_4) ont respectivement une durée de 7, 0 et 7 unités de temps, alors l'intervalle relatif de tir associé à la transition t_5 (à partir de l'état atteint lorsque t_5 devient sensibilisée) est $[MIN_5, MAX_5] = [\max(0, 0, 0), \min(2, 0, 0)] = [0, 0]$. Ainsi, la transition t_5 doit être tirée dès qu'elle est sensibilisée. De même en raisonnant à l'aide des intervalles absolus de validité temporelle (en supposant que le scénario débute au temps absolu $\tau=0$) et en utilisant l'opérateur de synchronisation « et » défini en section 2.5 on obtient un intervalle absolu de synchronisation égal à $[\max(7, 4, 7), \max(7, \min(9, 6, 9))] = [7, 7]$. Dans ce cas de figure l'unité de synchronisation (P_5, t_5) , en « avance » par rapport aux deux autres, sera retardée au point d'être temporellement incorrecte. Cependant dans ce cas la dérive temporelle induite sur l'unité de synchronisation « en avance » (P_5, t_5) est comprise entre $7-6=1$ et $7-4=3$ unités de temps.

Si nous adoptons un point de vue exécution et considérons que (P_4, t_5) , (P_5, t_5) et (P_6, t_5) se terminent respectivement aux dates absolues 9, 6 et 9, l'instant de synchronisation est alors égal à $\tau^{syn}=7$.

5.3.3 Règles de Tir pour une Transition de Type « ou-fort »

La sémantique de synchronisation de type « ou-fort » a pour objectif de garantir la correction temporelle de l'unité de synchronisation la plus en avance (*i.e.* celle atteignant au plus tôt la borne supérieure de son intervalle absolu de validité temporelle). Cette unité de synchronisation est définie dynamiquement en fonction des dérives temporelles subies par les unités de synchronisation « amont » dans le schéma de synchronisation inter-flux considéré. Ainsi ce type de synchronisation est susceptible de provoquer la terminaison ou « accélération » des unités de synchronisation « en retard ». La synchronisation de type « ou-fort » n'est donc pas en mesure de garantir la correction temporelle des arcs « en retard ».

Définition 4.4: Une transition t_k telle que $Syn(t_k)=ou\text{-}fort$, et $\forall a_i \in A_k^+, IDVT(a_i) = (\alpha_i, d_i, \beta_i)$, est tirable à l'instant relatif θ à partir de l'état $S=(M, I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$MIN_k = \min_i(\alpha_i) \leq \theta \leq \min_i(\beta_i) = MAX_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{fir} , associé à un point de synchronisation de type « ou-fort » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min \left\{ \tau \in H / \left(\exists i \in I_k, \left((\tau \geq \tau_i^{min}) \wedge (\tau \geq \tau_i^{fn}) \right) \vee \left(\tau \geq \tau_i^{max} \right) \right) \wedge \left(\tau \geq \tau^{low} \right) \right\}$$

Ce point de vue exécution de la synchronisation de type « ou-fort » assure la continuité de la fin d'exécution de l'unité de synchronisation la plus en avance avec le début des unités de synchronisation suivantes. D'autres

implantations de la synchronisation de type « ou-fort » tendant par exemple à minimiser la dérive temporelle des unités de synchronisation « en retard » pourraient être proposées.

Considérons à titre d'exemple le scénario de synchronisation de la figure 4.8, et associons à la transition t_5 une sémantique de synchronisation de type « ou-fort ». Si nous supposons que les unités de synchronisation (P_1, t_2) , (P_3, t_3) et (P_4, t_4) ont respectivement une durée de 5, 0 et 5 unités de temps⁶, alors l'intervalle relatif de tir associé à la transition t_5 (à partir de l'état atteint lorsque t_5 devient sensibilisée) est $[MIN_5, MAX_5]=[0,1]$. De même en raisonnant à l'aide des intervalles absolus de validité temporelle (en supposant que le scénario débute au temps absolu $\tau=0$) et en utilisant l'opérateur de synchronisation « ou-fort » défini en section 2.5 on obtient un intervalle absolu de synchronisation égal à $[\max(5, \min(5,4,5)), \max(5, \min(7,6,7))]=[5,6]$. Dans ce cas de figure l'unité de synchronisation (P_5, t_5) , en « avance » par rapport aux deux autres, sera temporellement correcte. Par contre, les deux unités de synchronisation en retard sont susceptibles de ne pas être exécutées complètement.

Si nous adoptons l'implantation du point de vue exécution proposée ci dessus et considérons que (P_4, t_5) , (P_5, t_5) et (P_6, t_5) se terminent respectivement aux dates absolues 7, 5 et 7, l'instant de synchronisation est alors égal à $\tau^{syn}=5$.

5.3.4 Règles de Tir pour une Transition de Type « ou »

La synchronisation de type « ou » introduit la notion d'indéterminisme de synchronisation. En effet, cette sémantique de synchronisation a pour objectif de garantir la cohérence temporelle ou la terminaison complète de l'une quelconque des unités de synchronisation impliquées dans le schéma de synchronisation considéré. Ainsi un point de synchronisation de type « ou » peut être atteint à tout instant ultérieur à τ^{min} et compris entre le minimum des bornes inférieures et le maximum des bornes supérieures des intervalles absolus de validité temporelle des unités de synchronisation considérées. L'indéterminisme de synchronisation intrinsèque à la synchronisation de type « ou » lui confère une capacité à intégrer des notions d'erreur ou de défaillance de synchronisation dans les schémas de synchronisation inter-flux. Cette sémantique de synchronisation peut induire (comme pour le type « ou-fort ») « l'accélération » des unités de synchronisation « en retard » ou bien (comme pour le « et-faible ») le « ralentissement des unités de synchronisation « en avance ». Il convient de noter que ce type de synchronisation n'implique la correction d'aucune unité de synchronisation spécifique: elle ne garantit seulement qu'une date au plus tôt et une date au plus tard pour le point de synchronisation.

Définition 4.5: Une transition t_k telle que $Syn(t_k)=ou$, et $\forall a_i \in A_k^+, IDVT(a_i) = (\alpha_i, d_i, \beta_i)$, est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

⁶ Pour ces dates de terminaison nous obtenons suite aux tirs des transitions t_2, t_3, t_4 du RdPFT de la Figure 4.8 l'état $S_3=(M_3, I_3)$ tel que $M_3=(p_4(1), p_5(1), p_6(1))$ et $I_3=((0,1,2), (0,0,1), (0,1,2))$. De même dans ce cas les intervalles absolus de validité temporelle des arcs (p_4, t_5) , (p_5, t_5) et (p_6, t_5) sont respectivement $(5,6,7)$, $(4,5,6)$ et $(5,6,7)$.

$$MIN_k = \min_i(\alpha_i) \leq \theta \leq \max_i(\beta_i) = MAX_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « ou » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min \left\{ \begin{array}{l} \tau \in H / \left(\exists j \in I_k, (\tau \geq \tau_j^{min}) \wedge (\tau \geq \tau_j^{fin}) \wedge (\tau \geq \tau^{roa}) \right) \\ \vee \\ (\forall i \in I_k, \tau \geq \tau_i^{max}) \end{array} \right\}$$

Ce point de vue exécution de la synchronisation de type « ou » est conduit par la première unité de synchronisation dont on détecte la terminaison et ayant dépassé la borne inférieure de son intervalle de validité temporelle. Ainsi la synchronisation de type « ou », en permettant à une unité de synchronisation de ne pas terminer dans son intervalle absolu de validité temporelle, a une capacité à modéliser des défaillances de synchronisation. Comme nous l'avons précisé antérieurement, une infinité d'autres implantations de cette sémantique de synchronisation peuvent être proposées.

Considérons à titre d'exemple le scénario de synchronisation de la figure 4.8, et associons à la transition t_5 une sémantique de synchronisation de type « ou ». Si nous supposons que les unités de synchronisation (P_1, t_2) , (P_3, t_3) et (P_4, t_4) ont duré respectivement 5, 0 et 5 unités de temps, alors l'intervalle relatif de tir associé à la transition t_5 (à partir de l'état atteint lorsque t_5 devient sensibilisée) est $[MIN_5, MAX_5] = [0, 2]$. De même, en raisonnant à l'aide des intervalles absolus de validité temporelle (en supposant que le scénario débute au temps absolu $\tau=0$) et en utilisant l'opérateur de synchronisation « ou » défini en section 2.5, on obtient un intervalle absolu de synchronisation égal à $[\max(5, \min(5, 4, 5)), \max(7, 6, 7)] = [5, 7]$. Dans ce cas de figure indéterminisme de synchronisation de type « ou » assurera la correction temporelle des trois unités de synchronisation (cas d'un point de synchronisation dans l'intervalle $[5, 6]$) ou bien seulement celle de (P_4, t_5) et (P_6, t_5) (cas d'un point de synchronisation dans l'intervalle $]6, 7[$)).

Si nous adoptons l'implantation du point de vue exécution proposée ci dessus et si nous considérons que les unités de synchronisation (P_4, t_5) et (P_5, t_5) défont, et que (P_6, t_5) se termine à la date absolue 7 alors l'instant de synchronisation est alors égal $\tau^{syn}=7$.

5.3.5 Règles de Tir pour une Transition de Type « maître »

La sémantique de synchronisation de type « maître » a pour objectif de satisfaire pour le mieux les contraintes temporelles d'une unité de synchronisation, qualifiée de « maîtresse », statiquement prédéterminée parmi l'ensemble des unités de synchronisation impliquées dans le schéma de synchronisation inter-flux considéré. Ce type de synchronisation est susceptible de provoquer l'interruption des unités de synchronisation « en retard », ainsi que le « ralentissement » des unités de synchronisation « en avance » par rapport à l'unité de synchronisation « maîtresse ». Ce type de synchronisation peut être utilisé lorsque la sémantique applicative dépend fortement de la cohérence temporelle d'une unité de synchronisation.

Définition 4.6: Une transition t_k telle que $\text{Syn}(t_k)=\text{maître}$, $\text{AM}(t_k)=a_m$ et $\text{IDVT}(a_m) = (\alpha_m, d_m, \beta_m)$, est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$\text{MIN}_k = \alpha_m \leq \theta \leq \beta_m = \text{MAX}_k$$

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « maître » peut être défini par l'exécutif de la façon suivante:

$$\tau^{\text{syn}} = \min \left\{ \tau \in H / \left(\left((\tau \geq \tau_m^{\text{min}}) \wedge (\tau \geq \tau_m^{\text{fin}}) \right) \vee (\tau \geq \tau_m^{\text{max}}) \right) \wedge (\tau \geq \tau^{\text{ou}}) \right\}$$

Ce point de vue exécution de la synchronisation de type « maître » a pour objectif d'assurer une stricte continuité temporelle entre la terminaison de l'unité de synchronisation « maîtresse » et les unités de synchronisation suivantes. Comme nous l'avons précisé antérieurement une infinité d'autres implantations de cette sémantique de synchronisation peuvent être proposées.

Considérons à titre d'exemple le scénario de synchronisation de la figure 4.8, et associons à la transition t_5 une sémantique de synchronisation de type « maître », avec l'arc (P_5, t_5) comme arc « maître ». Si nous supposons que les unités de synchronisation (P_1, t_2) , (P_3, t_3) et (P_4, t_4) ont des durées respectives de 5, 0 et 5 unités de temps, alors l'intervalle relatif de tir associé à la transition t_5 (à partir de l'état atteint lorsque t_5 devient sensibilisée) est $[\text{MIN}_5, \text{MAX}_5]=[0,1]$. De même, en raisonnant à l'aide des intervalles absolus de validité temporelle (en supposant que le scénario débute au temps absolu $\tau=0$) et en utilisant l'opérateur de synchronisation « maître » défini en section 2.5 on obtient un intervalle absolu de synchronisation égal à $[\max(4,5), \max(5,6)]=[5,6]$.

Si nous adoptons l'implantation du point de vue exécution proposée ci dessus et si nous considérons que (P_4, t_5) , (P_5, t_5) et (P_6, t_5) se terminent respectivement aux dates absolues 7, 5 et 7, l'instant de synchronisation est alors égal à $\tau^{\text{syn}}=5$.

Les 4 sémantiques de synchronisation, dérivées de la synchronisation de type « maître », que nous allons expliciter dans les quatre sections suivantes seront plus amplement justifiées et commentées dans le Chapitre 6 consacré à l'étude de la synchronisation hypermédia.

5.3.6 Règles de Tir pour une Transition de Type «et-maître»

La synchronisation de type « et-maître » combine les sémantiques de synchronisation « et » et « maître ». Ainsi cette sémantique de synchronisation a pour objectif d'assurer pour le mieux à la fois la correction temporelle d'une unité de synchronisation « maîtresse » ainsi que le fait que chacune des unités de synchronisation considérées ait atteint la borne inférieure de son intervalle de validité temporelle.

Définition 4.7: Une transition t_k telle que $\text{Syn}(t_k)=\text{et-maître}$, $\text{AM}(t_k)=a_m$, et $\forall a_i \in A_k^+, \text{IDVT}(a_i) = (\alpha_i, d_i, \beta_i)$ est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$\text{MIN}_k = \max_i(\alpha_i) \leq \theta \leq \max(\max_i(\alpha_i), \beta_m) = \text{MAX}_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « et-maître » peut être défini par l'exécutif de la façon suivante:

$$\tau^{\text{syn}} = \min\left\{\tau \in H / \left(\forall i \in I_k, (\tau \geq \tau_i^{\text{min}}) \wedge ((\tau \geq \tau_i^{\text{fin}}) \vee (\tau \geq \tau_m^{\text{max}}))\right)\right\}$$

Ce point de vue exécution de la synchronisation de type « et-maître » garantit que chaque unité de synchronisation atteint la borne minimum de son intervalle de validité temporelle, tout en assurant pour le mieux la correction temporelle de l'unité de synchronisation maîtresse.

5.3.7 Règles de Tir pour une Transition de Type «ou-maître»

La synchronisation de type «ou-maître» combine les sémantiques de synchronisation de type «ou» et «maître». Ainsi cette synchronisation introduit une forme d'indéterminisme de synchronisation borné supérieurement par la borne supérieure de l'intervalle de validité temporelle de l'unité de synchronisation maîtresse.

Définition 4.8: Une transition t_k telle que $\text{Syn}(t_k)=\text{ou-maître}$, $\text{AM}(t_k)=a_m$, et $\forall a_i \in A_k^+, \text{IDVT}(a_i) = (\alpha_i, d_i, \beta_i)$ est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$\text{MIN}_k = \min_i(\alpha_i) \leq \theta \leq \beta_m = \text{MAX}_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{fir} , associé à un point de synchronisation de type « et-maître » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min \left\{ \begin{array}{c} \tau \in H / \left(\exists j \in I_k, (\tau \geq \tau_j^{\min}) \wedge (\tau \geq \tau_j^{\max}) \wedge (\tau \geq \tau^{iou}) \right) \\ \vee \\ (\tau \geq \tau_m^{\max}) \wedge (\tau \geq \tau^{iou}) \end{array} \right\}$$

Ce point de vue exécution de la synchronisation de type « et-maître » garantit que le point de synchronisation est atteint dès qu'une unité de synchronisation se termine ou dès que l'unité de synchronisation « maîtresse » atteint la borne supérieure de son intervalle de validité temporelle. Comme nous l'avons précisé ultérieurement une infinité d'autres implantations de cette sémantique de synchronisation peut être proposée.

5.3.8 Règles de Tir pour une Transition de Type « maître-faible »

La synchronisation de type « maître-faible » résulte de la combinaison des synchronisations de type « maître » et « et-faible ». Ce type de synchronisation est complémentaire de la synchronisation de type « ou-maître » en effet elle induit un indéterminisme de synchronisation borné inférieurement par la borne inférieure de l'intervalle de validité temporelle de l'unité de synchronisation maîtresse.

Définition 4.9: Une transition t_k telle que $Syn(t_k)=\text{maître-faible}$, $AM(t_k)=a_m$, et $\forall a_i \in A_k^+$, $IDVT(a_i)=(\alpha_i, d_i, \beta_i)$ est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$MIN_k = \alpha_m \leq \theta \leq \max_i(\beta_i) = MAX_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{fir} , associé à un point de synchronisation de type « et-maître » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min \left\{ \begin{array}{c} \tau \in H / \left(\exists i \in I_k, (\tau \geq \tau_i^{\min}) \wedge (\tau \geq \tau_i^{\max}) \wedge (\tau \geq \tau^{iou}) \wedge (\tau \geq \tau_m^{\min}) \right) \\ \vee \\ (\forall i \in I_k, \tau \geq \tau_i^{\max}) \end{array} \right\}$$

Ce point de vue exécution de la synchronisation de type « et-maître » garantit que le point de synchronisation est atteint dès que la terminaison d'une unité de synchronisation a été détectée alors que toutes les unités de synchronisation considérées ont débuté et que l'unité de synchronisation « maîtresse » a atteint la borne inférieure de son intervalle de validité temporelle. Comme nous l'avons précisé antérieurement une infinité d'autres implantations de cette sémantique de synchronisation peut être proposée.

5.3.9 Règles de Tir pour une Transition de Type « maître-fort »

La synchronisation de type « ou-fort » résulte de la combinaison des synchronisations de type « maître » et « ou-fort ». Elle a pour objectif de satisfaire au mieux les contraintes temporelles de l'unité de synchronisation « maîtresse », ainsi que celles de l'unité de synchronisation la plus « en avance ».

Définition 4.10: Une transition t_k telle que $Syn(t_k)=\text{maître-fort}$, $AM(t_k)=a_m$, et $\forall a_i \in A_k^+$, $IDVT(a_i) = (\alpha_i, d_i, \beta_i)$ est tirable à l'instant relatif θ à partir de l'état $S=(M,I)$ si et seulement si les deux conditions suivantes sont satisfaites:

- la transition t_k est sensibilisée,
- l'instant relatif θ satisfait les inégalités suivantes:

$$MIN_k = \alpha_m \leq \theta \leq \max(\min_i(\beta_i), \alpha_m) = MAX_k$$

Avec i parcourant l'ensemble I_k .

Si nous considérons un point de vue exécution, l'instant précis de synchronisation, τ^{syn} , associé à un point de synchronisation de type « et-maître » peut être défini par l'exécutif de la façon suivante:

$$\tau^{syn} = \min\left\{\tau \in H / \left(\exists j \in I_k, (\tau \geq \tau_m^{min}) \wedge (\tau \geq \tau_j^{fin}) \wedge (\tau \geq \tau^{min})\right)\right\}$$

Ce point de vue exécution de la synchronisation de type « maître-fort » garantit que le point de synchronisation est atteint dès qu'une unité de synchronisation se termine et que l'unité de synchronisation maîtresse atteint la borne inférieure de son intervalle de validité temporelle. Comme nous l'avons précisé antérieurement une infinité d'autres implantations de cette sémantique de synchronisation peut être proposée.

5.3.10 Règles de Tir Globales pour une Transition dans un RdPFT

Les règles de tir précédemment formalisées considèrent, dans un état donné, une transition prise individuellement, et font donc abstraction de l'ensemble des autres transitions potentiellement tirables à partir de l'état considéré. Nous avons vu que ces règles de tir permettent d'associer un intervalle de tir $[MIN_k, MAX_k]$ à toute transition t_k sensibilisée dans un état $S=(M,I)$. De plus, cet intervalle est défini en fonction du type de synchronisation associé à la transition t_k considérée et des intervalles dynamiques de validité temporelle des arcs associés à t_k . En résumé, l'intervalle de tir, noté $[MIN_k, MAX_k]$, de la transition t_k sensibilisée dans l'état $S=(M,I)$ est obtenu de la façon suivante:

$$\text{si } (Syn(t_k) = \text{et}) \text{ alors } [MIN_k, MAX_k] = \left[\max_i(\alpha_i), \max(\min_i(\beta_i), \max_i(\alpha_i)) \right]$$

$$\text{si } (Syn(t_k) = \text{et - faible}) \text{ alors } [MIN_k, MAX_k] = \left[\max_i(\alpha_i), \max_i(\beta_i) \right]$$

$$\text{si } (Syn(t_k) = \text{ou}) \text{ alors } [MIN_k, MAX_k] = \left[\min_i(\alpha_i), \max_i(\beta_i) \right]$$

$$\text{si } (Syn(t_k) = \text{ou - fort}) \text{ alors } [MIN_k, MAX_k] = \left[\min_i(\alpha_i), \min_i(\beta_i) \right]$$

$$\begin{aligned}
& \text{si } (Syn(t_k) = \text{maître}) \wedge (MA(t_n) = a_n) \text{ alors } [MIN_k, MAX_k] = [\alpha_n, \beta_n] \\
& \text{si } (Syn(t_k) = \text{ou - maître}) \wedge (MA(t_n) = a_n) \text{ alors } [MIN_k, MAX_k] = [\min(\alpha_i), \beta_n] \\
& \text{si } (Syn(t_k) = \text{et - maître}) \wedge (MA(t_n) = a_n) \text{ alors } [MIN_k, MAX_k] = \left[\max_i(\alpha_i), \max(\beta_n, \max_i(\alpha_i)) \right] \\
& \text{si } (Syn(t_k) = \text{maître - fort}) \wedge (MA(t_n) = a_n) \text{ alors } [MIN_k, MAX_k] = [\alpha_n, \max(\min(\beta_i), \alpha_n)] \\
& \text{si } (Syn(t_k) = \text{maître - faible}) \wedge (MA(t_n) = a_n) \text{ alors } [MIN_k, MAX_k] = [\alpha_n, \max(\beta_i)]
\end{aligned}$$

avec l'indice i parcourant I_k et $\forall a_i \in A_k^+, IDVT(a_i) = (\alpha_i, d_i, \beta_i)$

Les règles de tir d'un RdPFT doivent considérer globalement l'ensemble des transitions sensibilisées dans un état donné. Ainsi ces règles de tir globales s'expriment de la façon suivante:

Définition 4.11: Une transition t_k est tirable à l'instant relatif θ depuis l'état $S=(M,I)$ si et seulement si les conditions suivantes sont satisfaites:

- La transition t_k est sensibilisée dans l'état $S=(M,I)$, autrement dit:

$$(\forall p \in P)(M(p) \geq \text{Pré}(p, t_k))$$

- L'instant relatif de tir θ vérifie les inégalités suivantes:

$$MIN_k \leq \theta \leq \min_n(MAX_n)$$

Avec n parcourant l'ensemble des indices des transitions sensibilisées dans l'état S .

La deuxième condition de tir impose que le tir d'une transition ne se fasse pas à une date telle que bien qu'appartenant à l'intervalle de tir de la transition considérée elle provoque le non respect de la sémantique de synchronisation d'une autre transition. Ainsi à l'instant relatif $\min_n(MAX_n)$ la transition ayant cette valeur comme borne supérieure d'intervalle de tir doit être tirée.

On remarquera que les intervalles de tir sont définis à partir d'instant relatifs et non d'instant absolus. Cela implique, selon un point de vue « modélisation », que chaque transition peut avoir une horloge locale à des fins de simulation. Cette caractéristique s'avère particulièrement intéressante selon un point de vue « exécution ». En effet, elle signifie que seules des horloges locales à chaque site sont nécessaires pour la gestion de schémas de synchronisation de type $n-1$ (i.e. n serveurs envoyant des flux devant être synchronisés sur un site client).

6. Propriétés des RdPFT

Nous avons montré que le modèle RdPFT possède un pouvoir de modélisation permettant de spécifier aisément des schémas de synchronisation inter-flux dans des systèmes faiblement synchrones. Ainsi, les RdPFT apportent, relativement aux modèles préexistants, une contribution indiscutable en terme de pouvoir de modélisation. Cependant, nous avons vu que la contribution d'un modèle formel s'exprime non seulement en fonction de son

pouvoir de modélisation mais aussi en fonction de ses pouvoirs d'expression et d'analyse. Ainsi l'objet de cette section est de mettre en évidence des propriétés démontrant que ce nouveau modèle possède des pouvoirs d'analyse et d'expression supérieurs ou égaux à ceux des modèles préexistants. En particulier, les réseaux de Petri Temporels (RdPT) constituent une excellente base de comparaison. En effet, ce modèle possède le pouvoir d'expression d'une machine de Turing. De plus de puissantes méthodes d'analyse [BD91] [BAB93] permettent de générer le graphe d'accessibilité d'un réseau de Petri temporel et ainsi d'en analyser les propriétés logiques et temporelles.

La propriété suivante montre que tout RdPT peut être associé à un RdPFT équivalent. De plus cette propriété montre que tout RdPT peut être facilement transformé en un RdPFT équivalent. La notion d'équivalence utilisée ici implique une équivalence de trace et une relation de bisimulation. C'est une équivalence forte qui résulte d'une équivalence entre le système de transition temporisé (*i.e.* le graphe d'états temporisé) généré à partir d'un RdPT, à l'aide de la notion d'état proposée dans [BD91], et celui créé à partir d'un RdPFT, à l'aide de la notion d'état d'un RdPFT introduite dans la section 4.1 de ce Chapitre.

Propriété 4.1: Soient un RdPT $R_1=(P_1, T_1, Pré_1, Post_1, M_0^1, IT)$ (avec IT la fonction associant à chaque transition un intervalle temporel), et un RdPFT $R_2=(P_2, T_2, Pré_2, Post_2, M_0^2, IVT, Syn, AM)$ vérifiant les propriétés suivantes:

$$\begin{aligned} P_1 &= P_2 \\ T_1 &= T_2 \\ Pré_1 &= Pré_2 \\ Post_1 &= Post_2 \\ IM_0^1 &= IM_0^2 \end{aligned}$$

$$\begin{aligned} \forall t_i \in T_1, IT(t) = (\alpha_i, \beta_i) &\Leftrightarrow \forall a_j \in A_i^+, IVT(a_j) = (x_j, n_j, y_j) = (\alpha_j, m_j, \beta_j) \\ \forall t \in T_2, Syn(t) &= et - faible \end{aligned}$$

Alors R_1 et R_2 sont équivalents au sens où toute séquence de tir réalisable à partir de l'état initial d'un réseau est réalisable à partir de l'état initial de l'autre réseau et aboutit à des états équivalents c'est à dire induisant les mêmes intervalles de tir pour les transitions sensibilisées.

Démonstration:

Avant de réaliser cette démonstration il convient de noter que l'état d'un RdPT, tel que défini par [BD91], est donné par un couple $S=(M, I)$ avec:

- M le marquage courant, et
- I l'ensemble des intervalles dynamiques des transitions sensibilisées.

De plus les règles d'évolution entre états font évoluer le marquage et l'ensemble I d'une façon conforme aux règles d'évolution des RdPFT en remplaçant la notion d'arc temporel par celle de transition temporelle.

Les notations suivantes seront utilisées dans cette démonstration:

- $S_{1,n} = (M_{1,n}, I_{1,n})R_1$ et $S_{2,n} = (M_{2,n}, I_{2,n})$ désignent respectivement le $n^{\text{ième}}$ état de R_1 et R_2 .
- $T_{1,n}^s$ et $T_{2,n}^s$ l'ensemble des transitions sensibilisées dans les états respectifs $S_{1,n}$ et $S_{2,n}$
- A_n^s désigne l'ensemble des arcs de R_2 sensibilisés dans l'état $S_{2,n}$
- IDT_n et $IDVT_n$ désignent respectivement les fonctions retournant l'intervalle dynamique de toute transition et tout arc sensibilisés dans les états respectifs $S_{1,n}$ et $S_{2,n}$. Les conventions suivantes seront utilisées pour désigner ces intervalles dynamiques: $\forall t_i \in T_{1,n}^s, IDT_n(t_i) = (\alpha_{i,n}, \beta_{i,n}), \forall a_i \in A_n^s, IDVT_n(a_i) = (x_{i,n}, n_{i,n}, y_{i,n})$.
On remarquera que $\forall t_i \in T_{1,0}^s, IDT(t_i) = (\alpha_{i,0}, \beta_{i,0}) = (\alpha_i, \beta_i) = IT(t_i)$, de même, $\forall a_i \in A_0^s, IDVT(a_i) = (x_{i,0}, n_{i,0}, y_{i,0}) = (x_i, n_i, y_i) = IVT(a_i)$

Nous allons démontrer cette propriété par induction.

Les états initiaux de $S_{1,0} = (M_{1,0}, I_{1,0})R_1$ et $S_{2,0} = (M_{2,0}, I_{2,0})$ de R_2 sont égaux, en effet $M_{1,0} = M_{2,0}$, de plus une transition t_n de R_2 est tirable à l'instant relatif θ à partir de $S_{2,0}$ si t_n est sensibilisée et si :

$$\forall a_i \in A_n^+ / IVT(a_i) = (x_i, n_i, y_i),$$

$$MIN_n = \min_i(x_i) \leq \theta \leq \min_k \left(\max_j(y_j^{(k)}) \right) = \min_k(MAX_k), i \in I_n, j \in I_k, t_k \in T_{0,2}^s$$

l'indice k parcourant l'ensemble des indices des transitions sensibilisées dans l'état courant. Les hypothèses initiales impliquent que $T_{1,0}^s = T_{2,0}^s$. De plus par hypothèse nous avons:

$$\forall t_n \in T_1 = T_2, \forall (a_i, a_j) \in A_n^+ \times A_n^+, IVT(a_i) = IVT(a_j) = (\alpha_n, m_n, \beta_n)$$

donc $\forall t_n \in T_{2,0}^s, \min_i(x_i) = \alpha_n$ et $\max_i(y_i) = \beta_n, i \in I_n$. Ce qui implique l'équivalence suivante:

$$\forall t_n \in T_{2,0}^s, \min_i(x_i) \leq \theta \leq \min_k \left(\max_j(y_j^{(k)}) \right), i \in I_n, j \in I_k, t_k \in T_{2,0}^s$$

$$\Leftrightarrow \forall t_n \in T_{1,0}^s, \alpha_n \leq \theta \leq \min_k(\beta_k), t_k \in T_{1,0}^s$$

Cette équivalence implique donc que les états $S_{1,0}$ et $S_{2,0}$ définissent le même ensemble de transitions tirables et induisent pour chaque transition tirable le même intervalle de tir.

Supposons que les deux réseaux sont dans deux états équivalents, notés $S_{1,n}$ et $S_{2,n}$ suite à une même séquence de tir. Ainsi, par définition de la notion d'équivalence d'états, les mêmes transitions sont tirables depuis $S_{1,n}$ et $S_{2,n}$ et ont des intervalles de tir identiques dans ces deux états. Autrement dit nous avons:

$$M_{1,n} = M_{2,n}$$

$$\forall t_n \in T_{1,n}^s = T_{2,n}^s / \left(IDT_n(t_k) = (\alpha_{k,n}, \beta_{k,n}) \right) \wedge \left(\forall i \in I_{t_n}, IDVT_n(a_i) = (x_{i,n}, n_{i,n}, y_{i,n}) \right), \\ \left(\alpha_{k,n} = \min_i(x_{i,n}) \right) \wedge \left(\beta_{k,n} = \max_i(y_{i,n}) \right) / i \in I_{t_n}$$

Soit t_r une transition tirable dans R_1 et R_2 à partir des états équivalents $S_{1,n}$ et $S_{2,n}$, et soit θ l'instant de tir relatif de t_r à partir de $S_{1,n}$ et $S_{2,n}$. D'après les règles d'évolution des états d'un RdPT et d'un RdPFT les états $S_{1,n} = (M_{1,n}, I_{1,n})$ et $S_{2,n} = (M_{2,n}, I_{2,n})$ évoluent vers deux états $S_{1,n+1} = (M_{1,n+1}, I_{1,n+1})$ et $S_{2,n+1} = (M_{2,n+1}, I_{2,n+1})$ définis comme il suit:

- $M_{1,n+1} = M_{2,n+1}$
- tout intervalle dynamique $(\alpha_{k,n}, \beta_{k,n})$ de $I_{1,n}$ associé à une transition t_k de R_1 sensibilisée dans $S_{1,n}$ et n'étant pas en conflit avec le tir de t_r devient $(\alpha_{k,n+1}, \beta_{k,n+1}) = (\max(0, \alpha_{k,n} - \theta), \max(0, \beta_{k,n} - \theta))$ dans $I_{1,n+1}$.
- tout intervalle dynamique $(x_{i,n}, n_{i,n}, y_{i,n})$ de $I_{2,n}$ associé à un arc a_i de R_2 sensibilisée dans $S_{2,n}$ et n'étant pas en conflit avec le tir de t_r devient $(x_{i,n+1}, n_{i,n+1}, y_{i,n+1}) = (\max(0, x_{i,n} - \theta), n_{i,n} - \theta, \max(0, y_{i,n} - \theta))$ dans $I_{2,n+1}$.
- l'intervalle de validité temporelle (α_k, β_k) de toute transition t_k nouvellement sensibilisée suite au tir de t_r est introduit dans $I_{1,n+1}$.
- l'intervalle de validité temporelle (x_i, n_i, y_i) de tout arc a_i nouvellement sensibilisé suite au tir de t_r est introduit dans $I_{2,n+1}$.

Ainsi l'intervalle de tir d'une transition t_k sensibilisée dans l'état $S_{1,n+1}$ de R_1 est égal à:

- $IT(t_k) = (\alpha_k, \beta_k)$ si t_k est nouvellement sensibilisée,
- $IDT_{n+1}(t_k) = (\alpha_{k,n+1}, \beta_{k,n+1}) = (\max(0, \alpha_{k,n} - \theta), \max(0, \beta_{k,n} - \theta))$ avec $IDT_n(t_k) = (\alpha_{k,n}, \beta_{k,n})$ sinon.

L'intervalle de tir associé à t_k dans l'état $S_{2,n+1}$ de R_2 est obtenu de la façon suivante:

$$\text{Si } t_k \text{ est nouvellement sensibilisée alors } \exists i \in I_k / IDVT_{n+1}(a_i) = IVT(a_i) = (x_i, n_i, y_i) = (\alpha_k, m_k, \beta_k), \text{ de plus} \\ \forall i \in I_{t_k}, (x_{i,n+1} \leq x_i = \alpha_k) \wedge (y_{i,n+1} \leq y_i = \beta_k) \\ \Rightarrow \left(\max_i(x_{i,n+1}) = \alpha_k \right) \wedge \left(\max_i(y_{i,n+1}) = \beta_k \right)$$

Donc dans ce cas l'intervalle de tir de t_k dans R_2 est égal à $IT(t_k) = (\alpha_k, \beta_k)$, et t_k a même intervalle de tir dans $S_{1,n+1}$ et $S_{2,n+1}$.

Si la transition t_k est sensibilisée dans $S_{2,n+1}$ et était sensibilisée dans $S_{2,n}$ tout en n'étant pas en conflit avec t_i alors: $\forall i \in I_k, IDVT(a_i) = (x_{i,n+1}, n_{i,n+1}, y_{i,n+1}) = (\max(0, x_{i,n} - \theta), n_{i,n} - \theta, \max(0, x_{i,n} - \theta))$, donc l'intervalle de tir de la transition t_k dans l'état $S_{2,n+1}$ est égal à:

$$\left(\max_i \left(\max(0, x_{i,n} - \theta) \right), \max_i (n_{i,n} - \theta), \max_i \left(\max(0, x_{i,n} - \theta) \right) \right) = \\ \left(\max \left(0, \max_i (x_{i,n}) - \theta \right), \max_i (n_{i,n}) - \theta, \max \left(0, \max_i (y_{i,n}) - \theta \right) \right)$$

or d'après l'hypothèse de récurrence, $\left(\alpha_{k,n} = \min_i (x_{i,n}) \right) \wedge \left(\beta_{k,n} = \max_i (y_{i,n}) \right) i \in I_k$, donc les états $S_{1,n+1}$ et $S_{2,n+1}$ associent le même intervalle de tir à la transition t_k .

Nous avons donc montré que les états $S_{1,n+1}$ et $S_{2,n+1}$ ont des marquages identiques et associent respectivement à toute transition sensibilisée de R_1 et R_2 des intervalles de tir identiques. Les états $S_{1,n+1}$ et $S_{2,n+1}$ sont donc équivalents.

Ceci achève la démonstration par récurrence et démontre ainsi l'équivalence du RdPT R_1 et du RdPFT R_2 . On remarquera de plus que cette propriété établit comment tout RdPT peut être traduit en un RdPAT équivalent.

CQFD.

Corollaire 4.1: Soient un réseau de Petri $R_1 = (P_1, T_1, Pr \acute{e}_1, Post_1, M_{1,0})$ et un réseau de Petri à flux temporel $R_2 = (P_2, T_2, Pr \acute{e}_2, Post_2, M_{2,0}, IVT, Syn, AM)$ vérifiant les propriétés suivantes:

$$\begin{aligned} P_1 &= P_2 \\ T_1 &= T_2 \\ Pr \acute{e}_1 &= Pr \acute{e}_2 \\ Post_1 &= Post_2 \\ M_{1,0} &= M_{2,0} \\ \forall a_i \in A_2^+, IVT(a_i) &= (0, n_i, +\infty) \\ \forall t_k \in T_2, Syn(t_k) &= et - faible \end{aligned}$$

Alors les réseaux R_1 et R_2 sont équivalents.

Démonstration:

Cette propriété résulte immédiatement du fait qu'un RdP $R_1 = (P_1, T_1, Pr \acute{e}_1, Post_1, M_{1,0})$ est équivalent à un RdPT

$R_3 = (P_3, T_3, Pr \acute{e}_3, Post_3, M_{3,0}, IT)$ tel que:

$$\begin{aligned} P_1 &= P_3 \\ T_1 &= T_3 \\ Pr \acute{e}_1 &= Pr \acute{e}_3 \\ Post_1 &= Post_3 \\ M_{1,0} &= M_{3,0} \\ \forall t_k \in T_3, IT(t_k) &= (0, +\infty) \end{aligned}$$

Or la propriété précédente établit l'équivalence de R_2 et de R_3 . Donc par transitivité R_1 et R_2 sont équivalents.
CQFD.

Propriété 4.2: Le modèle RdPFT a le pouvoir d'expression d'une machine de Turing.

Démonstration: Cette propriété résulte de la propriété 4.1. Etant donné que le modèle RdPT permet de réaliser le « test à zéro » ce modèle a le pouvoir d'expression d'une machine de Turing [Pet81]. Or tout RdPT peut être traduit en un RdPFT équivalent. Le modèle RdPFT a donc un pouvoir d'expression supérieur ou égal à celui du modèle RdPT. Ainsi le modèle RdPFT a le pouvoir d'expression d'une machine de Turing.

CQFD.

Propriété 4.3 : Les problèmes d'accessibilité et de « bornitude » sont indécidables pour les RdPFT.

Démonstration: Ce corollaire résulte du fait que tout RdPT peut être traduit sous la forme d'un RdPFT et que les problèmes d'accessibilité et de bornitude sont indécidables pour les RdPT [JLL77].

CQFD.

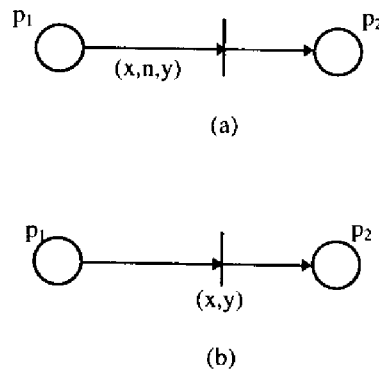


Figure 4.11: Traduction d'un schéma de synchronisation intra-flux d'un RdPFT (a) en un RDPT (b).

Nous avons vu comment tout RdPT peut être traduit sous forme d'un RdPFT équivalent. Ceci nous a permis de déduire des propriétés intéressantes quant au pouvoir d'expression du modèle RdPFT. Nous allons nous intéresser maintenant à la traduction inverse de RdPFT en RdPT. Cette opération de traduction est rendue a priori possible grâce au fait que le modèle RdPT a le pouvoir d'expression d'une Machine de Turing. L'objectif recherché à travers cette étape de traduction est double:

1. bénéficier pour le modèle RdPFT du pouvoir d'analyse des RdPT, et ainsi pouvoir appliquer aux RdPFT les méthodes d'analyse qui ont été proposées pour les RdPT [BD91] [BAB93].
2. montrer à travers la complexité du RdPT résultant de la traduction d'un RdPFT que le pouvoir de modélisation du modèle RdPFT est supérieur à celui du modèle RdPT.

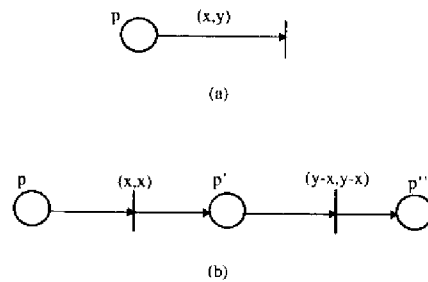


Figure 4.12 : Traduction d'une unité de synchronisation (a) sous la forme d'un RdPT (b) dans le cas d'un schéma de synchronisation inter-flux.

Propriété 4.4: Tout RdPFT peut être traduit sous la forme d'un RdPT.

Démonstration: La démonstration sera réalisée par construction. En effet nous allons montrer comment tout schéma de synchronisation d'un RdPFT peut être traduite en un RdPT induisant le même comportement temporel.

Tout point de synchronisation intra-flux d'un RdPFT (*i.e.* une transition avec un préfixe réduit à une seule place) peut être traduit sous la forme d'un RdPT selon la transformation immédiate illustrée par la figure 4.11.

Les figures 4.12-a et 4.12-b montrent comment un point de synchronisation inter-flux d'un RdPFT peut être traduit, en fonction du type de la transition, sous la forme d'un RdPT. Cette traduction consiste à mettre chaque unité de synchronisation d'un RdPFT sous la forme d'un RdPT qui explicite les événements fondamentaux participant à la définition d'une sémantique de synchronisation inter-flux; ces événements étant les dates d'entrée et de sortie des intervalles de validité temporelle des unités de synchronisation.

On traduit ainsi chaque unité de synchronisation en un RdPT explicitant les trois états pertinents d'une unité de synchronisation, à savoir respectivement quand une unité de synchronisation est avant, pendant et après son intervalle de validité temporelle. Les neuf sémantiques de synchronisation inter-flux exprimées sous la forme de RdPT s'obtiennent alors par de simples combinaisons temporelles de ces états.

On remarquera que cette traduction induit une équivalence entre le graphe d'état du RdPFT initial et une projection du graphe d'état associé au RdPT résultant de la traduction, la projection considérée s'attachant à extraire du graphe d'état du RdPT tous les marquages contenant les nouvelles places introduites suite à l'opération de traduction.

Tout schéma de synchronisation intra-flux et inter-flux d'un RdPFT pouvant être traduit sous la forme d'un RdPT (figures 4.13 et 4.14) nous en déduisons donc que tout RdPFT peut être traduit sous la forme d'un RdPT. Ainsi le modèle RdPFT hérite, par l'intermédiaire de la traduction vers le modèle RdPT, de l'ensemble des techniques d'analyse développées pour les RdPT [BD91][BAB93].

CQFD.

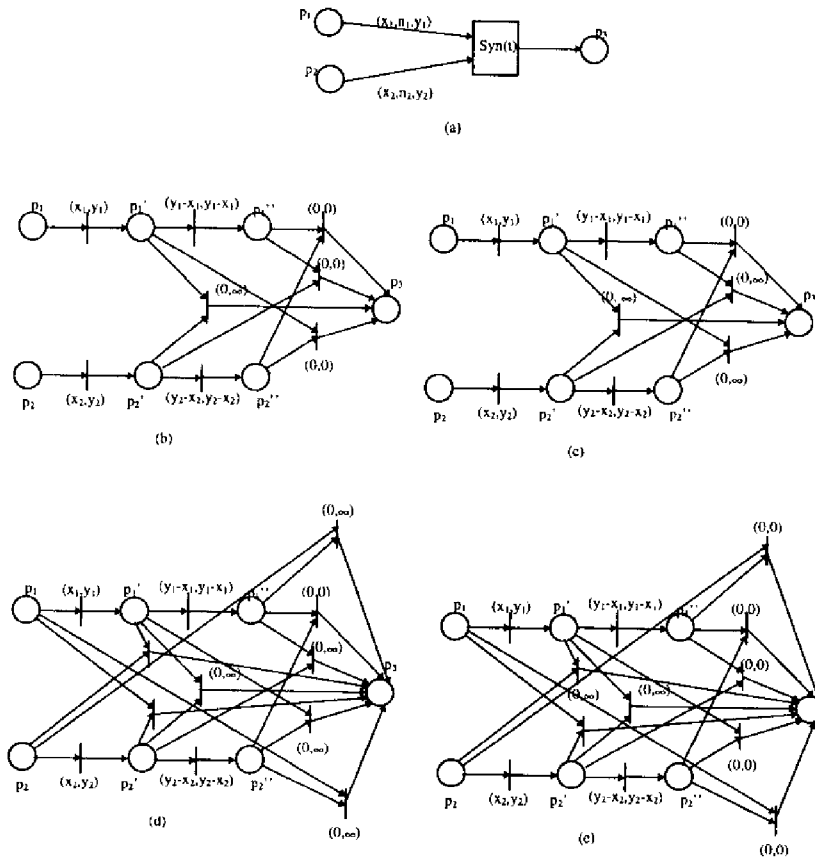


Figure 4.13 : Traduction d'un schéma de synchronisation inter-flux d'un RdPFT (a) sous forme d'un RdPT (b-c-d-e) en fonction du type de la synchronisation inter-flux. (b) transition de type « et », (c) transition de type « et-faible », (d) transition de type « ou », (e) transition de type « ou-fort ».

Sachant que tout RdPFT peut être traduit sous la forme d'un RdPT, se pose alors la question de la justification du modèle RdPFT. Nous avons vu que les pouvoirs d'expression et d'analyse des deux modèles sont identiques. Cependant le modèle RdPFT apporte une contribution en terme de pouvoir de modélisation. En effet, il apparaît qu'un schéma de synchronisation inter-flux exprimé sous la forme d'un RdPFT se traduit en un RdPT beaucoup plus complexe que le schéma initial. De plus cette traduction induit, en fonction du nombre d'unités de synchronisation considérées, une explosion combinatoire des places et des transitions résultantes. Ainsi le modèle RdPFT possède, pour la spécification des contraintes de synchronisation en environnement faiblement synchrone, un pouvoir de modélisation supérieur au modèle RdPT.

Lemme 4.1: Un RdPFT est borné (respectivement T-borné) si et seulement si le RdPT obtenu suite à l'opération de traduction est borné (respectivement T-borné).

Démonstration: On rappelle que par définition un réseau de Petri est T-borné s'il existe un entier k tel que aucune de ses transitions ne peut être sensibilisée plus de k fois par tout marquage accessible. Ce lemme résulte immédiatement du fait que, d'après les schémas de traduction introduits par les figures 4.12 et 4.13, tout schéma

de synchronisation élémentaire à m places d'entrées et à n places de sorties, exprimé sous la forme d'un RdPFT, est traduit en un RdPT vivant, conservatif, acyclique et ayant m places d'entrées et m places de sorties. *CQFD.*

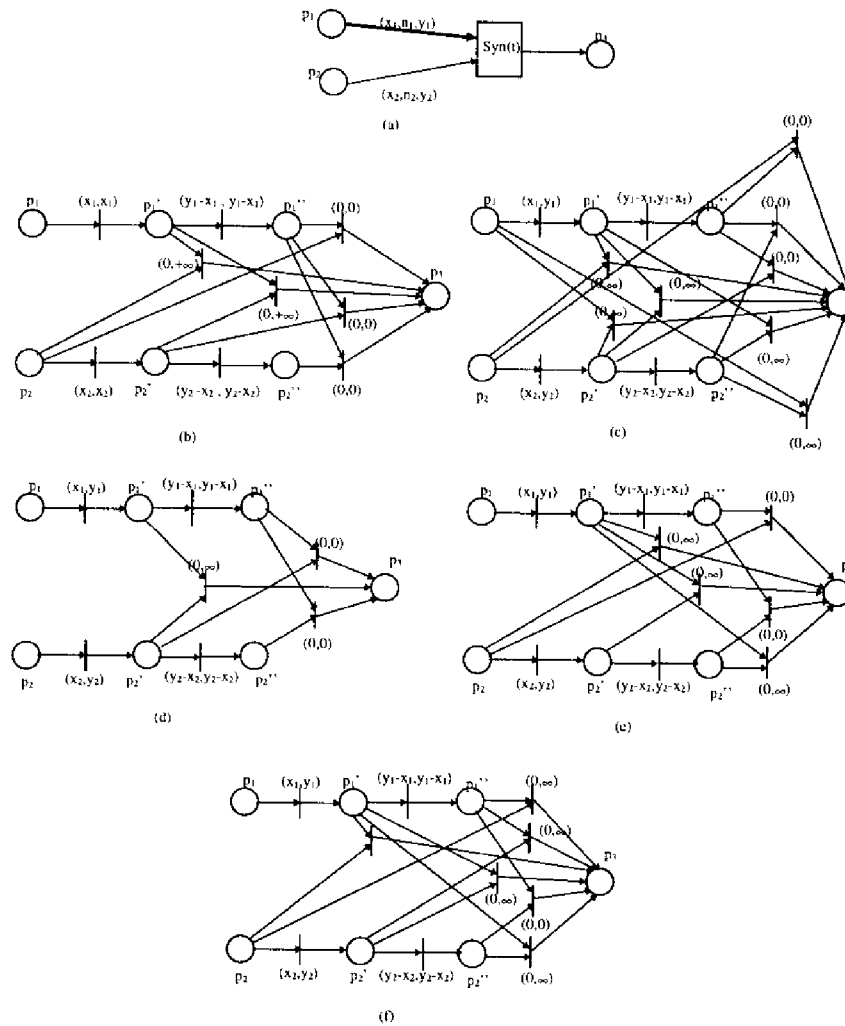


Figure 4.14 : Traduction d'un schéma de synchronisation inter-flux d'un RdPFT (a) sous forme d'un RdPT (b-c-d-e-f) en fonction du type de la synchronisation inter-flux. (b) transition de type « maître », (c) transition de type « ou-maître », (d) transition de type « et-maître », transition de type « maître-fort », (e) transition de type « maître-faible ».

Les RdPFT induisent un accroissement des contraintes de tir du réseau de Petri de base sous-jacent. Ainsi les conditions nécessaires et suffisantes de « bornitude » des RdP fournissent des conditions suffisantes de « bornitude » des RdPFT. Cependant ces conditions, par leur caractère faible, ont montré leurs limites pour la conception de protocoles à l'aide de RdPT [BD91]. Nous avons vu précédemment que la propriété de « bornitude » est indécidable pour les RdPFT. Cependant le lemme précédent nous permet avec l'aide des résultats développés dans [BD91] de mettre en évidence des conditions suffisantes de bornitude plus adaptées.

Les conditions de bornitude des RdPFT énoncées par la suite s'appuient sur la notion de classe d'états d'un RdPT, introduite dans [BD91]. Cette notion de classe d'états permet de réaliser une analyse énumérative des propriétés des RdPT. Le concept de classe d'états étend la notion d'état d'un RdPT, qui ne permet pas de réaliser une analyse énumérative et exhaustive des propriétés d'un RdPT. En effet si l'on considère une transition tirable à partir d'un état donné, étant donné le caractère dense des intervalles de tir, il existe une infinité de dates de tir potentielles de cette transition à partir de l'état considéré. Ainsi il existe une infinité d'états successeurs de l'état considéré. Par contre le concept de classe d'états permet de décrire sous une forme finie l'ensemble des états potentiellement atteignables suite au tir d'une transition à partir d'une classe d'état donnée. Une classe d'état peut donc être appréhendée comme une réunion éventuellement infinie d'états. Ainsi, à la différence du graphe d'états d'un RdPT le graphe de classes d'états d'un RdPT est un système de transition étiqueté non plus par des dates de tir de transitions mais par des intervalles de tir. Plus formellement une classe d'états d'un RdPT est décrite par un couple $C=(M,D)$ avec:

- M le marquage courant, et
- D le domaine de tir tel que $D = \{x / Ax \leq b\}$.

Le domaine de tir D est un système d'inéquations linéaires comprenant une variable, notée x_i , pour chaque transition t_i sensibilisée dans la classe d'état courante. Les contraintes associées dans D aux variables x_i sont de deux types, à savoir $\alpha_i \leq x_i \leq \beta_i$ et $x_i - x_j \leq \gamma_{i,j}$, et permettent d'associer, à partir de la classe d'états considérée, un intervalle de tir à chaque transition sensibilisée. Des règles d'évolution entre classes d'états permettent de calculer la nouvelle classe d'état obtenue suite au tir d'une transition. De plus [Men82] introduit un algorithme de mise sous forme canonique d'un domaine. Cet algorithme s'avère être fondamental pour l'analyse énumérative d'un RdPT, car pour que deux domaines soient égaux il faut et il suffit que leurs formes canoniques respectives soient égales. Suite à ce bref rappel de la notion de classe d'états d'un RdPT nous allons maintenant introduire des conditions « nécessaires et suffisantes » ainsi que des conditions suffisantes de « bornitude » d'un RdPFT. Ces propriétés résultent immédiatement de l'utilisation conjuguée du Lemme 4.1 et des conditions de « bornitude » d'un RdPT démontrées dans [BD91].

Propriété 4.5: Un RdPFT est borné si et seulement si son RdPT équivalent possède un nombre fini de classes d'états.

Démonstration: Cette propriété résulte du fait qu'un RdPT a un nombre fini de classes d'états si et seulement si est borné. Le lemme 4.1 permet alors d'en déduire immédiatement cette propriété.

CQFD.

Propriété 4.6: Un RdPFT T-borné est borné s'il n'existe pas de paires de classes d'états $C=(M,D)$ et $C'=(M',D')$, du RdPT équivalent, accessibles à partir de la classe d'état initiale et vérifiant:

1. C' est accessible à partir de C
2. $\forall p \in P M'(p) \geq M(p)$

Démonstration: Cette propriété résulte du lemme 4.1 et du théorème, introduit dans [BD91], spécifiant qu'un RdPT T-borné est borné s'il n'existe pas deux classes d'états vérifiant les conditions 1) et 2). En effet soient R un RdPFT et R' son RdPFT équivalent, et soit P la propriété « il n'existe aucune paire de classes d'états $C=(M,D)$ et $C'=(M',D')$, du RdPT équivalent, accessibles à partir de la classe d'état initiale vérifiant 1) et 2) ».

Nous avons:

$$P \Rightarrow (R' \text{ T-borné} \Rightarrow R' \text{ est borné})$$

or d'après le lemme 4.1, $(R \text{ borné} \Leftrightarrow R' \text{ borné}) \wedge (R \text{ T-borné} \Leftrightarrow R' \text{ T-borné})$ donc

$$P \Rightarrow (R \text{ T-borné} \Rightarrow R \text{ est borné})$$

CQFD.

7. Exemples

7.1 Un Schéma de « lip-synchronization »

7.1.1 Spécification Informelle du Problème

Le schéma de synchronisation continue considéré dans cette étude de cas met en jeu un flux audio et un flux vidéo, il reprend les spécifications de contraintes de « lip-synchronization » initialement données par [SHH92]. Dans un schéma de synchronisation continue de type « lip-synchronization » deux niveaux de synchronisation doivent être considérés:

- la synchronisation intra-flux permettant d'assurer le respect des contraintes temporelles portant sur les unités de synchronisation de chaque flux pris individuellement. Plus précisément, ce niveau définit pour chaque flux continu les intervalles de validité temporelle (*i.e.* les durées nominales et les giges admissibles) de chaque unité d'information.
- la synchronisation inter-flux relative à l'ensemble des flux considérés et spécifiant les dérives inter-flux admissibles.

Les contraintes de synchronisation intra-flux devant être vérifiées par chaque flux sont les suivantes:

- Les unités de synchronisation audio ont une durée nominale de 30ms et n'admettent pas de gigue. Autrement dit l'intervalle de validité temporelle de chaque unité de synchronisation vidéo est égal à (30,30,30).
- Les unités de synchronisation vidéo ont une durée nominale de 40 ms et admettent une gigue négative et positive de 5ms. L'intervalle de validité temporelle de chaque unité de synchronisation vidéo est donc égal à (35,40,45).

Les contraintes de synchronisation inter-flux devant être vérifiées par l'ensemble des flux sont les suivantes:

- Le flux vidéo peut être en avance d'au plus 15ms sur le flux audio,

- Le flux vidéo peut être en retard d'au plus 150 ms sur le flux audio.

7.1.2 Une Solution à Base du Modèle RdPFT

La solution illustrée par la figure 4.15 se propose de spécifier ce schéma de « lip-synchronization » à l'aide d'un schéma de synchronisation de type « série-parallèle ». L'avantage offert par une telle modélisation réside principalement dans la facilité avec laquelle des schémas de synchronisation inter-flux peuvent être spécifiés. En effet les contraintes de synchronisation inter-flux, conformément au paradigme de la « ligne temporelle », sont spécifiées par simple fusion de transitions entre le flux audio et le flux vidéo. L'hypothèse de continuité, satisfaite par les deux flux, nécessite d'appliquer des points de synchronisation entre des sous-flux ayant les mêmes durées nominales. Autrement dit les points de synchronisation entre le flux audio et le flux vidéo doivent être appliqués entre des sous flux ayant des durées nominales égales à un multiple du Plus Petit Commun Multiple des durées nominales des unités de synchronisation audio et vidéo (*i.e.* 120 unités de temps). La granularité de synchronisation est alors déterminée par les dérives inter-flux admissibles. L'avance maximum admissible pour le flux vidéo par rapport au flux audio est de 15ms ; sachant que le flux audio n'est soumis à aucune dérive, cette avance résulte uniquement de l'effet cumulatif de la gigue du flux vidéo. La gigue négative des unités de synchronisation vidéo étant de 5 unités de temps, une dérive négative de 15ms peut donc être atteinte suite au traitement de trois unités de synchronisation (correspondant au traitement de 4 unités de synchronisation audio).

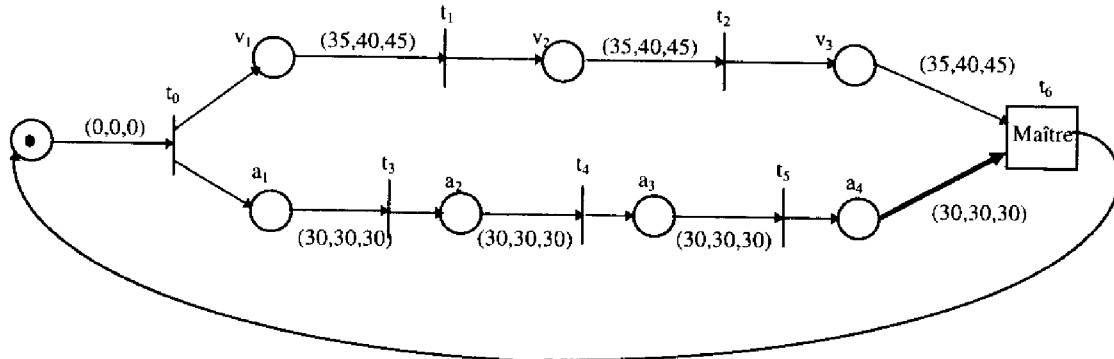


Figure 4.15: Spécification d'un schéma de « lip-synchronization » à l'aide du modèle RdPFT

Ainsi de façon à éviter une avance supérieure à 15ms du flux vidéo par rapport au flux audio, des points de synchronisation doivent être appliqués entre le sous flux composé de 3 unités de synchronisation vidéo et le sous flux audio correspondant (composé de 4 unités de synchronisation). Le flux audio n'admettant aucune dérive, les transitions de synchronisation inter-flux seront de type « maître » avec le flux audio comme flux maître.

Les limitations de cette modélisation résident dans le fait que la granularité de synchronisation doit être choisie en fonction de la plus petite dérive admissible (*i.e.* le minimum de l'avance (*i.e.* 15ms) et du retard maximum (*i.e.* 150 ms), c'est à dire dans cet exemple 15ms). Cette contrainte induit généralement des schémas de synchronisation proposant une dérive plus restrictive que souhaitée. Ainsi le schéma de synchronisation illustré

par la figure 4.15 permet d'obtenir, conformément aux spécifications, une avance maximum de la vidéo par rapport à l'audio de 15ms. Par contre ce schéma de synchronisation induit un retard maximum (*i.e.* une dérive positive) de moins de 15ms, et non 150ms, entre le flux audio et le flux vidéo.

7.2 Une Application Multimédia

Considérons le RdPFT représenté par la figure 4.16. Cette spécification modélise le scénario de présentation multimédia du Département d'Informatique d'une Grande Ecole. Cinq flux d'informations multimédias sont mis en jeu dans ce scénario, à savoir:

- un flux vidéo, modélisé par la séquence d'unités de synchronisation $s_1=((V_1,t_3),(V_2,t_4),(V_3,t_5))$ et présenté dans la fenêtre 1 (figure 4.17). Ce flux vidéo présente, à l'aide de la séquence V_1 , l'historique et la localisation du département Informatique; puis les différentes équipes de recherche et leurs principaux projets sont introduits par la séquence vidéo V_2 ; finalement la séquence V_3 conclut par un bilan prospectif prévoyant un brillant futur pour le département.
- un flux audio, modélisé par la séquence d'unités de synchronisation $s_2=((A_1,t_2),(A_2,t_3),(A_3,t_4),(A_4,t_5),(A_5,t_{10}))$ et associé à la fenêtre 1 (figure 4.17). Ce flux vidéo est constitué d'un discours d'introduction (A_1), du commentaire audio relatif au flux vidéo (respectivement A_2 , A_3 et A_4), et d'un discours de conclusion (A_5).
- un flux textuel, modélisé par la séquence $s_3=((T_2,t_9),(T_3,t_{10}))$, et présenté dans la fenêtre 2 (figure 4.17). Ces deux présentations textuelles décrivent les principales étapes de l'évolution du Département Informatique (texte T_2), ainsi que les différents groupes et leurs principaux projets (texte T_3). Ce flux contient deux commentaires textuels relatifs aux flux audio et vidéo.
- un flux d'images fixes, modélisé par la séquence $s_4=((I_1,t_6),(I_3,t_7),(I_4,t_8),(I_5,t_{10}))$, et présenté dans la fenêtre 3 (figure 4.17). Ces images contiennent le logo du département (image I_1), une mosaïque d'images représentant les chercheurs confirmés du département (image I_3), une mappemonde localisant les partenaires internationaux du département (image I_4) et finalement un nouvelle présentation du logo du département (image I_5).
- un flux hétérogène, c'est à dire composé d'une image, d'une animation de synthèse et d'un texte. Ce flux est modélisé par la séquence $s_5=((I_2,t_6),(S_1,t_7),(T_1,t_8),(I_6,t_{10}))$ et présenté dans la fenêtre 4 (figure 4.17). Ce flux hétérogène présente une vue panoramique du département (image I_2), une animation de synthèse illustrant les principaux thèmes de recherche (animation synthétique S_1), un texte détaillant la liste des principaux partenaires internationaux du département (texte T_1), et une photographie de tout le personnel du département (image I_6).

Dans ce scénario de synchronisation, des points de synchronisation inter-flux sont mis en oeuvre entre les flux audio et vidéo, le flux d'images et le flux hétérogène, et les flux de texte, d'images et audio.

Les points de synchronisation placés entre les flux audio et vidéo sont modélisés par les transitions t_3 , t_4 et t_5 . Ces points de synchronisation sont appliqués périodiquement toutes les 30 unités de temps entre ces deux flux

continus afin d'éviter une dérive inter-flux excessive. Etant donné que le flux audio a des contraintes de synchronisation plus rigoureuses que le flux vidéo, ces points de synchronisation inter-flux sont de type « maître » et les unités de synchronisation audio sont définies comme arcs « maîtres ». Ces schémas de synchronisation inter-flux garantissent les dérives temporelles maximales suivantes entre les deux flux:

- $\max(|135-20|, |135-40|) = 15$, si le flux audio se termine à sa date de terminaison au plus tard,
- $\max(|25-20|, |25-40|) = 15$, si le flux audio se termine à sa date de terminaison au plus tôt.

Ces schémas de synchronisation garantissent dans tous les cas le respect des contraintes temporelles et la continuité des unités de synchronisation audio.

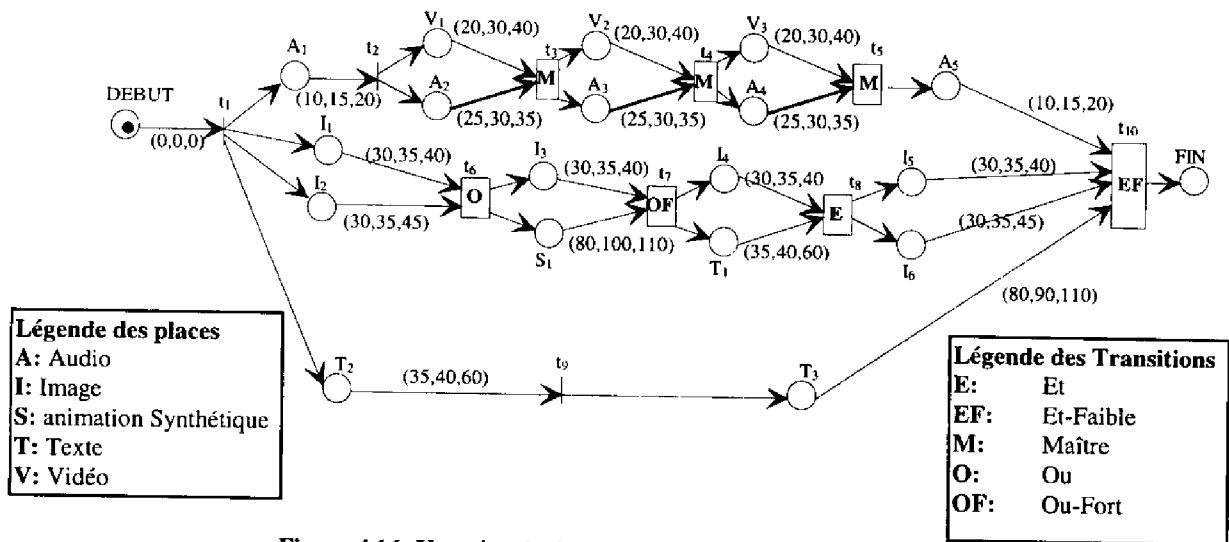


Figure 4.16: Un scénario de synchronisation multimédia

Le premier point de synchronisation inter-flux entre le flux d'images et le flux hétérogène est appliqué par l'intermédiaire de la transition t_6 de type « ou » entre les unités de synchronisation (I_1, t_6) et (I_2, t_6) . Conformément à la sémantique de synchronisation de type « ou », ce point de synchronisation est atteint si au moins une des deux unités de synchronisation se termine. Ainsi, l'utilisation d'une telle sémantique de synchronisation présuppose que la présentation complète d'au moins un des deux médias est suffisante pour la sémantique applicative. Ce point de synchronisation exprime une forme d'indéterminisme de synchronisation et peut donner lieu à ce titre à différentes implantations. Dans tous les cas, le point de synchronisation est dans l'intervalle temporel relatif $[\min(30,30), \max(40,45)]$. De plus ce point de synchronisation assure une stricte continuité entre la fin de I_1 ou I_2 et la présentation des unités d'information suivantes (*i.e.* (I_3, t_7) and (S_1, t_7)). Le second point de synchronisation inter-flux met en jeu une synchronisation de type « ou-fort » entre deux unités d'informations (*i.e.* (I_3, t_7) et (S_1, t_7)) telles que la durée minimum de présentation de l'animation de synthèse S_1 (*i.e.* 80) est beaucoup plus longue que la durée maximum de présentation de l'image fixe I_3 (*i.e.* 40). Ainsi l'animation de synthèse sera dans tous les cas interrompue avant la fin afin d'assurer le respect de la cohérence

temporelle de l'image. L'animation de synthèse sera donc présentée pendant une durée égale à la durée de présentation de l'image. Dans l'hypothèse la plus favorable (respectivement la plus défavorable) seules les 40 (respectivement 30) premières unités de temps de l'animation seront présentées. Ce type de synchronisation peut être utilisé si l'on suppose que seule la présentation complète de l'image est nécessaire pour la sémantique applicative. La présentation parallèle de I_4 et T_4 fait suite à celle de I_3 et S_1 . Ces présentations sont synchronisées par la transition t_8 de type « et ». Ce point de synchronisation satisfait dans tous les cas la relation $IAVT(I_4, t_8) \cap IAVT(T_4, t_8) = IVT(I_4, t_8) \cap IVT(T_4, t_8) \neq \emptyset$. Cette relation combinée au type « et » de la transition t_8 implique que la sémantique temporelle des deux unités de synchronisation sera toujours satisfaite.

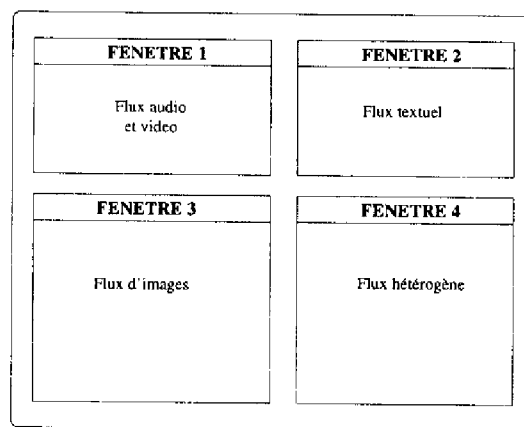


Figure 4.17 : Présentation spatiale du scénario multimédia

Finalement un point de synchronisation entre les flux audio, images fixes et texte est réalisé à la fin du scénario de synchronisation. Ce point de synchronisation est modélisé par la transition t_{10} , de type « et-faible », qui synchronise les présentations de A_5 , I_5 , I_6 et T_3 . Le type « et-faible » garantit l'exécution complète des 4 unités de synchronisation impliquées dans ce point de synchronisation. Ce point de synchronisation a été choisi car la sémantique applicative est conditionnée par la présentation complète des quatre médias. Ainsi l'utilisateur aura le temps d'entendre complètement le commentaire audio, de voir les images fixes, et de lire le texte. Ces quatre présentations peuvent être activées à des dates différentes en fonction des dérives temporelles subies par les parties « amont » de ces quatre flux.

Le comportement dynamique de ce scénario de synchronisation peut être simulé à partir de la notion d'état et des règles d'évolution entre états introduites dans la section 4.2. Ainsi l'évolution de ce scénario à partir de l'état initial peut être décrit par la séquence d'états suivante:

L'état initial de ce RdPFT est décrit par le couple $S_0=(M_0, I_0)$ avec:

- $M_0=(DEBUT(1))$
- $I_0=((0,0,0))$

La transition t_1 est la seule transition tirable à partir de l'état initial. Cette transition doit être tirée, à partir de S_0 , à l'instant $\theta_0 = 0$. Nous obtenons alors un nouvel état $S_1=(M_1,I_1)$ tel que:

- $M_1 = (A_1(1), I_1(1), I_2(1), T_2(1))$ où (A_1, t_2) , (I_1, t_6) (I_2, t_6) et (T_2, t_9) sont les arcs sensibilisés,
- $I_1 = ((10, 15, 20), (30, 35, 40), (30, 35, 45), (35, 40, 60))$

Seules les transitions t_2 , t_6 et t_9 sont sensibilisées dans l'état S_1 . Les règles de tir des RdPFT nous permettent d'obtenir pour chaque transition prise individuellement les intervalles de tir suivants:

- $[10, 20]$ pour t_2 ,
- $[\min(30, 30), \max(40, 45)]$ pour la transition t_6 de type « ou »,
- $[35, 60]$ pour t_9

Les conditions de tir globales pour un ensemble de transitions telles que définies dans la section 4.3.10 conduisent à associer aux transitions sensibilisées les intervalles de tir suivants:

- $[10, \min(\max(40, 45), 20, 60)] = [10, 20]$ pour t_2
- $[\min(30, 30), \min(\max(40, 45), 20, 60)] = \emptyset$ pour t_6 et,
- $[35, \min(\max(40, 45), 20, 60)] = \emptyset$ pour t_9 .

Notons que t_6 et t_9 ne sont pas tirables depuis l'état S_1 . Si t_2 est tirée depuis S_1 l'instant relatif $\theta_1 = 20$, nous obtenons alors le nouvel état $S_2=(M_2,I_2)$ tel que:

- $M_2 = (A_2(1), I_1(1), I_2(1), T_2(1), V_1(1))$ où t_3 , t_6 et t_9 sont les seules transitions sensibilisées
- $I_2 = (25, 30, 35)$ pour (A_2, t_3) ,
- $(\max(0, 30 - \theta_1), 35 - \theta_1, \max(0, (40 - \theta_1))) = (10, 15, 20)$ pour (I_1, t_6) ,
- $(\max(0, 30 - \theta_1), 35 - \theta_1, \max(0, (45 - \theta_1))) = (10, 15, 25)$ pour (I_2, t_6) ,
- $(\max(0, 35 - \theta_1), 40 - \theta_1, \max(0, (60 - \theta_1))) = (15, 20, 40)$ pour (T_2, t_9) ,
- $(20, 30, 40)$ pour (V_1, t_2)

Les intervalles de tir des transitions sensibilisées sont alors:

- $[25, \min(\max(20, 25), 35, 40)] = \{25\}$ pour la transition t_3 de type « maître » t (i.e. la transition t_3 ne peut être tirée à partir de S_2 qu'à l'instant relatif 25).
- $[\min(10, 10), \min(\max(20, 25), 35, 40)] = [10, 25]$ pour la transition de type « ou » t_6 .
- $[15, \min(\max(20, 25), 35, 40)] = [15, 25]$ pour la transition t_9 .

Si t_6 est tirée à $\theta_2=25$ (à cette date l'unité de synchronisation (I_2, t_6) est temporellement correcte alors que (I_1, t_6) ne l'est pas), nous atteignons alors l'état $S_3=(M_3, I_3)$ tel que:

- $M_3=(A_2(1), I_3(1), S_2(1), T_2(1), V_1(1))$
- $I_3 = ((0,5,10), (30,35,40), (80,100,110), (0,-5,15), (0,5,15))$.

Les intervalles de tir des transitions sensibilisées dans S_3 sont alors:

- $[0, \min(10, \min(40, 110), 15)] = [0, 10]$ pour la transition t_3 de type « maître »,
- $[30, \min(10, \min(40, 110), 15)] = \emptyset$ pour la transition t_7 de type « ou-fort »,
- $[0, \min(10, \min(40, 110), 15)] = [0, 10]$ pour la transition t_9 .

Si t_3 est tirée depuis S_3 à l'instant relatif $\theta_3=5$, nous atteignons alors l'état $S_4=(M_4, I_4)$ tel que:

- $M_4=(A_3(1), I_3(1), S_1(1), T_2(1), V_2(1))$
- $I_4 = ((25,30,35), (25,30,35), (75,95,105), (0,-10,10), (20,30,40))$.

La séquence de tir suivante peut être réalisée à partir de l'état S_4 :

Le tir de t_9 à l'instant relatif $\theta_4=10$ conduit à l'état $S_5=(M_5, I_5)$ tel que:

- $M_5=(A_3(1), I_3(1), S_1(1), T_3(1), V_2(1))$
- $I_5 = ((15,20,25), (15,20,25), (65,85,95), (80,90,110), (10,20,30))$.

Le tir de t_7 à l'instant $\theta_5=15$ à partir de l'état S_5 conduit à l'état $S_6=(M_6, I_6)$ tel que:

- $M_6=(A_3(1), I_4(1), T_1(1), T_3(1), V_2(1))$
- $I_6 = ((0,5,10), (30,35,40), (35,40,60), (65,75,95), (0,5,15))$.

Le tir de t_4 à l'instant $\theta_6=0$ à partir de l'état S_6 conduit à l'état $S_7=(M_7, I_7)$ tel que:

- $M_7=(A_4(1), I_4(1), T_1(1), T_3(1), V_3(1))$
- $I_7 = ((25,30,35), (30,35,40), (35,40,60), (65,75,95), (20,30,40))$.

Le tir de t_5 à l'instant $\theta_7=25$ à partir de l'état S_7 conduit à l'état $S_8=(M_8, I_8)$ tel que:

- $M_8=(A_5(1), I_4(1), T_1(1), T_3(1))$
- $I_8 = ((10,15,20), (5,10,15), (10,15,35), (40,50,70))$.

Le tir de t_8 à l'instant $\theta_8=21$ à partir de l'état S_8 conduit à l'état $S_9=(M_9, I_9)$ tel que:

- $M_9=(A_5(1), I_5(1), I_6(1), T_3(1))$

- $I_9 = ((0,3,8), (30,35,40), (30,35,45), (28,38,58))$.

Le tir de la transition t_{10} de type « et-faible » conduit à l'état final S_{10} . Conformément aux règles de tir de type « et-faible », l'intervalle de tir de la transition t_{10} à partir de S_{10} est égal à $[\max(0,30,28), \max(8,40,45,58)] = [30,58]$. Ainsi, si t_{10} est tirée à l'instant relatif $\theta_9 = 58$ alors les contraintes temporelles des unités de synchronisation (A_5, t_{10}) , (I_5, t_{10}) et (I_6, t_{10}) ne sont pas satisfaites.

8. Conclusion

Nous avons introduit dans ce chapitre un nouveau modèle formel appelé Réseaux de Petri à flux temporels. Ce modèle offre à la fois, à la différence des modèles disponibles jusqu'alors, de bons pouvoirs de modélisation, d'expression, et d'analyse des contraintes de synchronisation dans les systèmes faiblement synchrones (figure 4.18). En effet nous avons montré dans ce chapitre que le modèle RdPFT:

- possède le pouvoir d'expression d'une machine de Turing,
- permet de spécifier facilement des schémas de synchronisation inter-flux par simple fusion de transitions, conformément au paradigme de la « régie de production »,
- hérite par l'intermédiaire de possibilités de traduction dans le formalisme des Réseaux de Petri Temporels des techniques d'analyse développées pour ce modèle.

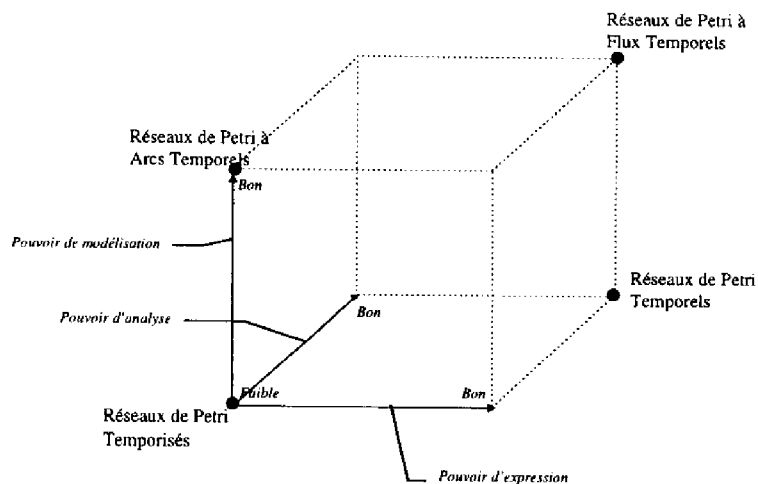


Figure 4.18 : Contribution du modèle RdPFT par rapport aux modèles préexistants

En proposant un modèle temporel basé sur un temps relatif et en utilisant des intervalles comme unités temporelles, le modèle RdPFT répond aux caractéristiques générales devant être offertes par un modèle formel pour la synchronisation multimédia (voir Chapitre 3). De plus, le Chapitre 3 a mis en évidence qu'un modèle pour la synchronisation multimédia doit offrir un pouvoir d'expression permettant :

- de modéliser des scénarios de synchronisation de complexité arbitrairement élevée,
- de modéliser l'indéterminisme temporel des unités de synchronisation,
- d'introduire de l'indéterminisme de synchronisation dans des schémas de synchronisation inter-flux et
- de modéliser des comportements adaptatifs aux ressources disponibles.

Il apparaît immédiatement que le pouvoir d'expression des RdPFT satisfait les deux premiers critères. De plus, l'indéterminisme de synchronisation peut être exprimé à l'aide de l'indéterminisme temporel des unités de synchronisation impliquées dans un schéma de synchronisation inter-flux. En effet, considérons à titre d'exemple, le RdPFT illustré par la figure 4.19. Ce RdPFT met en jeu une unité de synchronisation modélisant un délai pur (*i.e.* l'arc (T, t_1)) et deux unités de synchronisation modélisant des traitements multimédias (*i.e.* respectivement (P_2, t_2) et (P_3, t_2)). Si nous faisons abstraction de la gigue des unités de synchronisation, autrement dit si nous considérons exclusivement les durées nominales des unités de synchronisation, et si nous supposons que $(d \leq n_1) \wedge (d + n_2 \geq n_1)$, alors ce schéma de synchronisation modélise, conformément à l'algèbre temporelle de Allen (figure 3.2), une relation temporelle de type « P_1 recouvre P_2 ». Si nous faisons intervenir l'indéterminisme temporel des unités de synchronisation, en particulier si nous supposons que l'unité de synchronisation modélisant le délai a un intervalle de validité temporelle $(d_1, d, d_2) = (0, d, +\infty)$, alors ce RdPAT spécifie une disjonction de schémas de synchronisation déterministes, à savoir:

$$(P_1 \text{ recouvre } P_2) \vee (P_1 \text{ puis } P_2) \vee (P_1 \text{ avant } P_2) \vee (P_1 \text{ commence-avec } P_2)$$

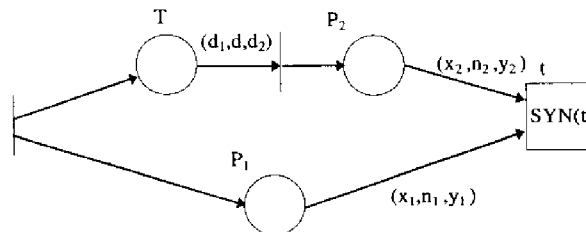


Figure 4.19: RdPFT modélisant un schéma de synchronisation potentiellement non-déterministe.

Ainsi le modèle RdPFT permet de spécifier l'un quelconque des 2^{13} schémas de synchronisation non-déterministes susceptibles d'être dérivés des 13 opérateurs de synchronisation déterministes définis par [All83]. En particulier le modèle RdPAT permet d'exprimer les 29 relations temporelles fondamentales non-déterministes⁷ mise en évidence par [WR94]. Cependant il est important de noter que l'indéterminisme de synchronisation intrinsèque aux schémas de synchronisation exprimés selon le formalisme des RdPFT peut être contrôlé à l'aide des 9 sémantiques de synchronisation introduites par le modèle RdPFT.

⁷ Ces 29 relations temporelles non-déterministes constituent un sous-ensemble des 2^{13} relations temporelles non-déterministes pouvant être exprimées entre deux intervalles temporels.

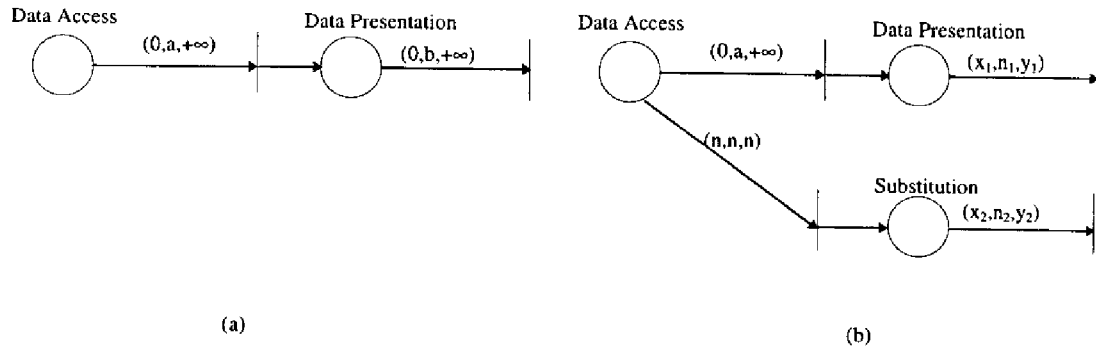


Figure 4.20: Modification d'un comportement « least effort » (a) en un comportement « best effort » (b).

Des comportements adaptatifs peuvent être spécifiés à l'aide du modèle RdPFT en exploitant la capacité des réseaux de Petri à modéliser des comportements non-déterministes. La figure 4.20 illustre, à titre d'exemple, la façon dont le comportement asynchrone d'un système multimédia distribué tel que le WWW peut être transformé en un comportement faiblement synchrone. Ainsi le RdPFT de la figure 4.20-a spécifie le comportement générique d'une application multimédia de type client-serveur pour laquelle toute opération de présentation nécessite au préalable une opération d'accès à des données distribuées. L'utilisateur d'une telle application peut observer sur un système distribué asynchrone des délais d'accès ou de présentation non bornés. Ce comportement asynchrone est modélisé sur la figure 4.20-a par une structure de synchronisation séquentielle composée de deux unités de synchronisation, modélisant respectivement les opérations d'accès et de présentation, associées à des intervalles de validité temporelle non-bornés. Ce comportement de type « le moins d'effort possible » (*i.e.* « least effort ») en matière de contraintes de synchronisation peut être remplacé par un comportement de type « le meilleur effort » (*i.e.* « best-effort ») spécifié par le RdPFT de la figure 4.20-b. Ce RdPFT spécifie que si l'opération d'accès peut être réalisée en au plus n unités de temps, alors l'objet multimédia est présenté à l'utilisateur. Sinon un traitement de remplacement, traduisant un comportement adaptatif, est réalisé. Ce traitement de substitution consiste généralement en un traitement « plus léger » et donc moins exposé à des gignages temporelles intempestives. Ainsi, à titre d'exemple, le traitement de substitution peut accéder à une ressource locale ou plus facilement accessible tout en assurant une sémantique de remplacement minimum par rapport à la ressource non disponible (e.g. une image fixe ou un texte peuvent remplacer une vidéo). De plus on peut noter que les opérations de présentation sont modélisées dans le RdPFT de la figure 4.20-b à l'aide d'unités de synchronisation associées à des intervalles de validité temporelle bornés. Ainsi, en fonction de ces contraintes temporelles, un exécutif de synchronisation peut appliquer des actions correctives ou réaliser des délais d'attentes conditionnels [Ste90] afin de garantir la sémantique temporelle des tâches de présentation, qui sont par définition locales au client.

Cet exemple montre comment le comportement dynamique d'un scénario de synchronisation peut s'adapter aux ressources disponibles. De plus, le pouvoir d'expression des RdPFT permet de modéliser les trois approches fondamentales pour le traitement des défaillances en matière de synchronisation [LG91a], à savoir: l'expansion (*i.e.* chaque unité de synchronisation est traitée même si elle est en retard), l'abandon (*i.e.* l'unité de

synchronisation en retard est sautée), la compensation (*i.e.* l'unité de synchronisation en retard est remplacée par un traitement de substitution).

Ainsi le modèle RdPFT apporte une contribution indiscutable au formalisme des réseaux de Petri étendus par le temps. Mais la contribution de ce nouveau modèle ne s'arrête pas là. En effet:

- le concept de système faiblement synchrone exploité par ce modèle permet de couvrir tout le spectre allant des systèmes purement synchrones aux systèmes purement asynchrones. Le caractère générique de ce modèle est renforcé par l'introduction de règles de synchronisation permettant de spécifier de façon abstraite et générale des contraintes de synchronisation mettant en jeu plusieurs unités de synchronisation, et par là même de recouvrir de façon très concise à partir d'une même spécification plusieurs mises en oeuvre potentielles d'un schéma de synchronisation inter-flux.
- le modèle RdPFT est, à notre connaissance, le premier modèle ayant introduit une sémantique formelle pour la notion de synchronisation en environnement faiblement synchrone. Ce modèle fournit de plus une classification complète des sémantiques de synchronisation en environnement faiblement synchrone. Le caractère « best effort » de ces sémantiques de synchronisation permet d'assurer dans tous les cas la continuité des flux de traitements, et par là même permet de garantir la « robustesse » des schémas de synchronisation.

Cependant ce modèle présente les deux limitations intrinsèques suivantes:

- Des risques intempestifs de désynchronisation, bien que pris en charge par le caractère « best-effort » des règles de synchronisation, sont susceptibles d'induire le non respect des contraintes temporelles de l'unité de synchronisation que l'on cherche à privilégier dans un schéma de synchronisation inter-flux. Le seul moyen d'empêcher ce paradoxe de synchronisation serait de fournir une technique de vérification mettant en évidence les schémas de synchronisation inter-flux exposés à de tels risques ; le spécifieur pouvant alors modifier les schémas de synchronisation mis en cause. Cette nécessité de disposer de telles techniques de vérification a été mise en évidence dans le Chapitre 3 où il est apparu qu'un modèle temporel basé sur un temps relatif et utilisant des intervalles comme unités temporelles doit offrir des techniques de vérification permettant de mettre en évidence les incohérences quantitatives et qualitatives des schémas de synchronisation inter-flux. Malheureusement de telles techniques ne peuvent être mises en oeuvre avec une complexité polynomiale dans le cas de RdPFT de structure et de complexité arbitraires. En effet [Wal83] a mis en évidence, dans le cadre des RdPAT, que les risques de désynchronisation intempestifs entre les arcs temporels impliqués dans un point de synchronisation ne peuvent pas, dans le cas général, être détectés à l'aide d'algorithmes de complexité polynomiale.
- Les règles de tir des RdP imposent la sensibilisation d'une transition comme condition nécessaire de tir. Ainsi ces règles traditionnelles ne permettent pas de modéliser la stratégie traditionnelle de resynchronisation consistant lorsqu'une dérive inter-flux excessive est observée à « accélérer » le flux en retard en « sautant » un certain nombre de ses unités de synchronisation. Autrement dit les RdPFT ne permettent de modéliser que des stratégies de resynchronisation inter-flux du type « attente ou expansion » (*i.e.* le flux en avance attend le flux en retard), et « saut ou abandon » d'une portée limitée à une unité de synchronisation, à savoir l'unité de synchronisation impliquée dans le point de synchronisation considéré. Plus généralement les RdPFT ne

permettent pas de modéliser des événements asynchrones, ou interruptions, susceptibles d'interrompre des scénarios de synchronisation de complexité arbitraire. Cette capacité d'interruption de scénarios de synchronisation s'avère fondamentale dans le contexte des systèmes hypermédias où l'activation de liens hypertextuels s'apparente à un événement asynchrone ou interruption.

Les deux chapitres suivants vont respectivement introduire une restriction et une extension du modèle RdPFT levant ces deux limitations.

Chapitre 5

Les Réseaux de Petri Structurés à Flux Temporels

1. Introduction

Nous avons mis en évidence dans le chapitre précédent que les sémantiques de synchronisation introduites par le modèle RdPFT sont susceptibles d'introduire des paradoxes temporels. En effet, étant donné la nécessité de satisfaire la condition de sensibilisation d'une transition en préalable à son tir, les règles de tir des RdPFT, à l'exclusion du « et », du « et-faible », et du « et-maître », ne sont pas en mesure de garantir systématiquement le respect des contraintes temporelles de l'unité de synchronisation que l'on cherche à privilégier. Autrement dit, l'unité de synchronisation dynamiquement ou statiquement privilégiée peut être retardée au point de ne plus satisfaire ses contraintes temporelles. Ce paradoxe temporel peut apparaître rédhibitoire pour des applications « temps-réel dur » qui ne sauraient se satisfaire de règles de synchronisation de type « best-effort ». Cependant, des sémantiques de synchronisation plus restrictives, garantissant dans tous les cas la sémantique temporelle de l'unité de synchronisation que l'on cherche à privilégier induiraient, à l'identique des problèmes intrinsèques aux RdPAT, des risques potentiels de comportement indéfini ou de non continuité des flux. Risques d'autant plus présents que, comme souligné dans le chapitre précédent, l'utilisateur peut décider de ne pas vouloir payer le prix d'algorithmes visant à mettre en évidence dans un RdPFT les risques de désynchronisation intempestifs, et qui sont dans le cas général de complexité non-polynomiale. Faut-il en déduire pour autant que le modèle RdPFT n'est pas adapté au temps réel critique ? L'objet de ce chapitre est de montrer qu'il n'en est rien, et qu'une sous-classe des RdPFT nommée Réseaux de Petri Structurés à Flux Temporels (RdPSFT) permet de mettre en oeuvre des techniques de vérification temporelles de complexité polynomiale. Ces techniques de vérification temporelle permettent ainsi de mettre en évidence, et en réponse de modifier, les schémas de synchronisation exposés à des risques excessifs de désynchronisation.

2. Organisation du Chapitre

Après avoir introduit le modèle RdPSFT nous montrons en quoi ce modèle est bien adapté à la modélisation de schémas de synchronisation mettant en jeu des flux continus de traitements. En particulier nous montrons comment les structures de synchronisation informellement introduites dans le second chapitre trouvent une expression formelle à l'aide des RdPSFT. Nous montrons ensuite comment ces structures de synchronisation peuvent être réduites de façon à leur appliquer des techniques de vérifications temporelles. Puis cette technique de réduction-vérification sera étendue à des RdPSFT de complexité arbitraire. Les propriétés algébriques de cette technique de réduction seront alors étudiées. Ces techniques de réduction permettent d'introduire formellement la notion de structure logique d'un système multimédia. Finalement la dernière partie de ce chapitre est consacrée à l'étude d'un exemple.

3. Les Réseaux de Petri Structurés à Flux Temporels

Le modèle OCPN, introduit dans le chapitre 3, permet de spécifier des scénarios de synchronisation multimédia de complexité arbitraire, à l'aide d'une approche structurée et minimaliste en terme du nombre de structures de synchronisation utilisées. Nous avons mis en évidence précédemment les limitations de ce modèle, limitations tenant en particulier à l'incapacité des OCPN à exprimer la variabilité temporelle des traitements. Cette limitation rédhibitoire peut être levée en transformant un OCPN, noté R , en RdPFT R' comme il suit:

1. Pour toute place P_i de durée n_i , appartenant au préfixe d'une transition¹ t_j , remplacer la durée nominale n_i par l'intervalle de validité temporelle (x_i, n_i, y_i) spécifiant la variabilité temporelle associée au traitement modélisé dans R par P_i . Associer cet intervalle de validité temporelle à l'arc (P_i, t_j) .
2. Suite à la substitution de toutes les durées nominales associées aux places de R par des intervalles de validité temporelle associés aux arcs issus de ces places, le RdPFT R' est obtenu en associant à chaque transition, gérant un point de synchronisation inter-flux, une règle de tir prise dans l'ensemble des 9 sémantiques de synchronisation introduites par le modèle RdPFT.

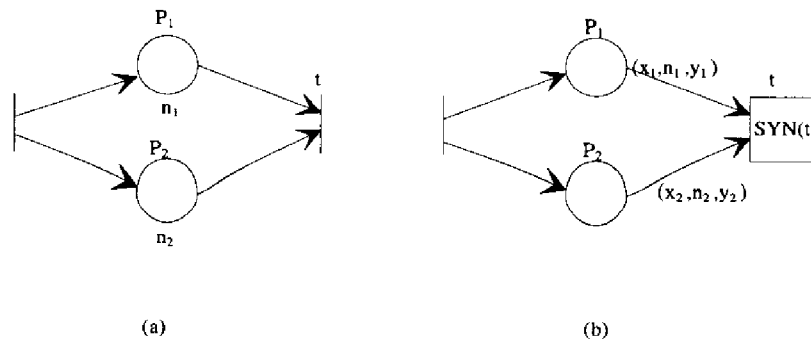


Figure 5.1: Traduction d'un OCPN en RdPSFT. (a) un schéma de synchronisation du type « P_1 commence avec P_2 », (b) la traduction de ce schéma sous la forme d'un RdPSFT.

¹Noter que cette transition est unique étant donnée la structure de graphe marqué d'un OCPN.

On notera que la première étape de cette transformation, en introduisant des intervalles de validité temporelle dans les relations temporelles de Allen, est susceptible de bouleverser la sémantique de ces schémas de synchronisation. Ainsi par exemple un schéma de type « P_1 commence avec P_2 » peut devenir « P_1 égale P_2 » ou bien « P_2 commence avec P_1 » (Figure 5.1). La deuxième étape de la traduction, permet de contrôler l'indéterminisme de synchronisation introduit par la première étape, en associant une sémantique de synchronisation à chaque point de synchronisation inter-flux. Ainsi, la deuxième étape est susceptible d'assurer un comportement de synchronisation déterministe.

Définition 5.1: Le RdPFT obtenu suite à l'application à un OCPN des deux étapes de transformation précédemment décrites est appelé Réseau de Petri Structuré à Flux Temporels (RdPSFT).

Les RdPSFT possèdent toutes les « bonnes propriétés » structurelles des OCPN (voir chapitre 3).

4. Structures de Synchronisation et RdPSFT

Nous allons montrer dans cette section que les RdPSFT permettent une formalisation des principaux concepts informellement introduits dans le second chapitre. En particulier nous allons définir à partir du formalisme des RdPSFT les principales structures de synchronisation, des plus élémentaires aux plus complexes, susceptibles d'intervenir dans un scénario de synchronisation. Il apparaîtra ainsi que les RdPSFT fournissent un bon support formel pour mettre en oeuvre le paradigme de composition de la « régie de production » à l'aide d'une approche minimaliste et structurée.

4.1 Structures de Synchronisation Élémentaires

4.1.1 Les Unités de Synchronisation

La notion d'unité de synchronisation telle qu'introduite dans le chapitre 2 modélise une tâche dédiée au traitement de une ou plusieurs unités d'information multimédias. Cette notion trouve une correspondance formelle dans le modèle RdPFT sous la forme d'un arc et de son intervalle de validité temporelle. Cette structure de synchronisation élémentaire constitue la structure de synchronisation de base pour la construction de scénarios de synchronisation.

Définition 5.2: On appelle *unité de synchronisation* d'un RdPFT $R=(P,T,Pré,Post, M_0, IVT, Syn, AM)$ tout élément appartenant à l'ensemble $A^+ = \{(p, t) \in P \times T / Pré(p, t) \neq 0\}$

La notion de gigue d'une unité de synchronisation est alors formellement définie de la façon suivante:

Définition 5.3: La *gigue d'une unité de synchronisation* est donnée par la fonction g telle que:

$$\begin{aligned} g: A &\rightarrow \mathcal{Q}^- \times \mathcal{Q}^+ \\ \forall a \in A^+ / IVT(a) &= (x, n, y) \\ g(a) &= (x - n, y - n) \end{aligned}$$

Le premier et le deuxième élément du couple spécifiant la gigue d'une unité de synchronisation sont appelés respectivement *gigue négative* et *gigue positive* de l'unité de synchronisation.

Définition 5.4 : La durée d'une unité de synchronisation est spécifiée par la fonction d définie de la façon suivante:

$$d: A^+ \rightarrow \mathcal{Q}^+$$

$$\forall a_i \in A^+, d(a_i) = \tau_i^{syn} - \tau_i^{deb}$$

La durée donnée par la fonction d tient compte du « contrôle » opéré par la sémantique de synchronisation associée au point de synchronisation considéré. La valeur retournée par cette fonction est à opposer à la valeur $\tau_i^{fin} - \tau_i^{deb}$ qui représente la durée de l'unité de synchronisation en l'absence d'une sémantique de synchronisation faiblement synchrone.

Définition 5.5: Une unité de synchronisation, notée a , est *temporellement correcte* si et seulement si $d(a) \in IVT(a)$.

On remarquera que la sémantique de synchronisation introduite par le modèle RdPFT garantit la correction temporelle des schémas de synchronisation intra-flux. De plus, dans le cas d'un schéma de synchronisation inter-flux, la sémantique de synchronisation des RdPFT, préserve la correction temporelle d'au moins une unité de synchronisation statiquement ou dynamiquement définie.

Définition 5.6: La *dérive temporelle* d'une unité de synchronisation est donnée par la fonction δ définie comme il suit:

$$\delta: A^+ \rightarrow \mathcal{Q}$$

$$\forall a \in A^+ / IVT(a) = (x, n, y), \delta(a) = d(a) - n$$

Cette définition induit celles d'avance et de retard d'une unité de synchronisation. En effet, une unité de synchronisation, notée a , sera dite:

- en avance si et seulement si $\delta(a) < 0$,
- en retard si et seulement si $\delta(a) > 0$
- à l'heure si et seulement si $\delta(a) = 0$

Considérons à titre d'exemple l'unité de synchronisation, notée a , représentée par la Figure 5.2.

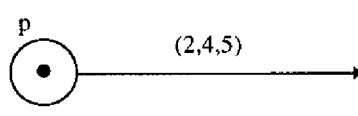


Figure 5.2: Une unité de synchronisation

Si nous supposons que la transition t est tirée à l'instant $\tau^{syn}=3$ (en supposant que $\tau^{deb}=0$), alors cette structure élémentaire de synchronisation est caractérisable par les paramètres donnés dans le tableau 5.1. On remarquera en particulier que l'unité de synchronisation est temporellement correcte et qu'elle est en avance d'une unité de temps par rapport à sa durée nominale.

Intervalle de validité temporelle de a	IVT(a)=(2,4,5)
Gigue de a	$g(a)=(-2,1)$
Durée de a	$d(a)=3$
Dérive de a	$\delta(a)= -1$

Tableau 5.1: Caractérisation temporelle de l'unité de synchronisation de la Figure 5.2

4.2 Structures Temporelles Composées

4.2.1 Les Flux

Les flux ont été définis dans le chapitre 2 comme étant des séquences d'unités de synchronisation. Ces structures de synchronisation explicitent les contraintes de synchronisation intra-flux. La notion de flux trouve une correspondance formelle immédiate dans le modèle RdPFT sous la forme de structures de synchronisation obtenues par composition récursive d'unités de synchronisation et/ou de flux à l'aide des relations temporelles séquentielles de Allen (Figure 3.2 a et b). On notera par le caractère « ; » la relation de séquentialité sur l'ensemble des arcs temporels d'un RdPFT ainsi définie:

$$\forall R \in RdPFT, \forall (a_i, a_j) \in A^2 / a_k = (p_k, t_k), k = \{i, j\}, a_i; a_j \Leftrightarrow p_j \in t_i^*$$

On remarquera que la relation de séquentialité ainsi définie est une relation d'ordre strict.

Définition 5.7: Un flux de longueur n est un n -uplet d'unités de synchronisation $f = (a_1, \dots, a_n) \in (A^+)^n$ tel que:

$$\forall i \in \{1, \dots, n-1\}, a_i; a_{i+1}, \text{ ce qui sera noté } a_1; \dots; a_n$$

Un flux $f = (a_1, \dots, a_n) / \forall i \in (1, \dots, n), a_i = (p_i, t_i)$ sera qualifié d'*élémentaire* si et seulement si:

$$\left(\forall j = 1, \dots, n-1, |^*t_j| = |t_j^*| = 1 \right)$$

Les ensembles des flux et des flux élémentaires d'un RdPFT seront respectivement notés F et F^e .

On remarquera qu'un flux est défini de façon unique et non ambiguë par une suite d'arcs entrant sur des transitions d'un RdPFT. Ainsi un flux ne constitue pas rigoureusement, au sens de la théorie des graphes, un chemin dans un RdPFT. Cependant la donnée d'un flux dans un RdPFT définit de façon unique un chemin, au sein du réseau considéré.

L'opérateur « ; » sur A peut être étendu sur l'ensemble F de la façon suivante:

$$\begin{aligned} \forall (f_1, f_2) \in F^2 / f_1 = (a_1, \dots, a_n), f_2 = (b_1, \dots, b_m) \\ f_1 ; f_2 \Leftrightarrow a_n ; b_1 \end{aligned}$$

Les intervalles de validité temporelle associés aux unités de synchronisation composant un faisceau induisent la notion d'intervalle de validité temporelle d'un faisceau.

Définition 5.8: L'intervalle de validité temporelle d'un flux est donné par la fonction IVT_f définie de la façon suivante:

$$\begin{aligned} IVT_f : S \rightarrow Q^+ \times Q^+ \times Q^+ \\ \forall f = (a_1, \dots, a_n) \in F / \forall a_i \in f, IVT(a_i) = (x_i, n_i, y_i) \\ IVT_f(f) = \sum_{i=1}^n IVT(a_i) = \left(\sum_{i=1}^n x_i, \sum_{i=1}^n n_i, \sum_{i=1}^n y_i \right) \end{aligned}$$

L'intervalle de validité temporelle d'un faisceau élémentaire représente le comportement temporel global d'un faisceau élémentaire.

Propriété 5.1: Tout flux élémentaire peut être réduit en un arc temporel unique à l'aide de la fonction de réduction r_f suivante:

$$\begin{aligned} r_f : F \rightarrow A^+ \\ \forall f = (a_1, \dots, a_n) \in F / IVT_f(f) = (x, n, y), \\ r_f(f) = a, IVT(a) = (x, n, y) \end{aligned}$$

Démonstration: L'existence et l'unicité de la forme réduite d'un flux élémentaire résulte immédiatement de la définition de la fonction de réduction de flux élémentaire. En effet,

$$\forall f = (a_1, \dots, a_n) \in F^e / IVT(a_i) = (x_i, n_i, y_i), \exists! (x, n, y) \in Q^+ \times Q^+ \times Q^+ / (x, n, y) = \left(\sum_{i=1}^n x_i, \sum_{i=1}^n n_i, \sum_{i=1}^n y_i \right)$$

Cette fonction de réduction permet de décrire sous une forme abstraite, par l'intermédiaire d'une unité de synchronisation, le comportement temporel d'un flux élémentaire.

CQFD.

Les fonctions définissant la gigue, la durée et la dérive temporelle d'une unité de synchronisation induisent respectivement celles définissant les notions de gigue, de dérive et de durée d'un flux.

Définition 5.9: La gigue d'un flux est donnée par la fonction g_f suivante:

$$\begin{aligned} g_f : F \rightarrow Q^+ \times Q^- \\ \forall f = (a_1, \dots, a_n) \in F / (IVT_f(f) = (x, n, y)) \wedge (\forall a_i, g(a_i) = (x_i - n_i, y_i - n_i)), \\ g_f(f) = (x - n, y - n) = \left(\sum_{i=1}^n x_i - \sum_{i=1}^n n_i, \sum_{i=1}^n y_i - \sum_{i=1}^n n_i \right) = \left(\sum_{i=1}^n (x_i - n_i), \sum_{i=1}^n (y_i - n_i) \right) = \sum_{i=1}^n g(a_i) \end{aligned}$$

La gigue d'un flux est donc la « somme » des giges des unités de synchronisation le constituant.

Définition 5.10: La durée d'un flux est définie par la fonction d_f telle que:

$$d_f : F \rightarrow \mathbb{Q}^+$$

$$\forall f = (a_1, \dots, a_n) \in F, d_f(f) = \sum_{i=1}^n d(a_i)$$

Définition 5.11: La dérive d'un flux est définie par la fonction δ_f telle que:

$$\delta_f : F \rightarrow \mathbb{Q}$$

$$\forall f = (a_1, \dots, a_n) \in F, \delta_f(f) = \sum_{i=1}^n \delta(a_i) = \sum_{i=1}^n (d(a_i) - n_i) = \sum_{i=1}^n d(a_i) - \sum_{i=1}^n n_i = d_f(f) - n$$

Définition 5.12: Un flux f est temporellement correct si et seulement si chacune des unités de synchronisation le composant est temporellement correcte.

On remarquera que si un flux f est temporellement correct alors $d_f(f) \in IVT_f(f)$, la réciproque n'étant pas vraie.

Considérons le flux élémentaire de longueur 3, noté f , représenté par la figure 5.3, et supposons que les durées des unités de synchronisation composant ce flux répondent aux caractéristiques temporelles suivantes: $d(p_1, t_1)=12$, $d(p_2, t_2)=8$, $d(p_3, t_3)=31$.

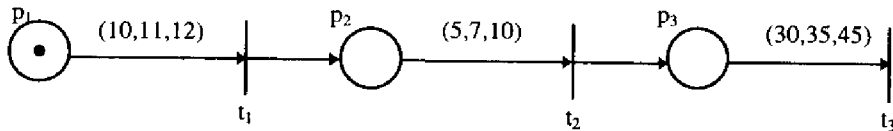


Figure 5.3: Modélisation d'un flux à l'aide d'un RdPFT

La caractérisation du comportement temporel de ce flux est alors décrite par les paramètres donnés par le tableau 5.2.

Intervalle de validité temporelle du flux f	$IVT_f(f)=(45,53,67)$
Gigue du flux f	$j_f(f)=(-8,14)$
Durée du flux f	$d_f(f)=51$
Dérive du flux f	$\delta_f(f)=-2$

Tableau 5.2: Caractérisation temporelle du flux représenté par la Figure 5.3

On remarquera de plus que le flux f est temporellement correct étant donné que les trois unités de synchronisation le constituant sont temporellement correctes.

4.2.2 Les Faisceaux

Les faisceaux sont des structures de synchronisation permettant de spécifier des contraintes de synchronisation inter-flux. Ainsi un faisceau résulte de la composition récursive de flux et de faisceaux à partir des relations temporelles de Allen mettant en oeuvre du parallélisme (Figure 3.2 c-g). Nous représenterons par le caractère « | » la relation de parallélisme sur l'ensemble des arcs temporels ainsi définie:

$$\forall R, \forall (a_1, a_2) \in (A^+), a_1 | a_2 \Leftrightarrow \exists (t_\alpha, t_\omega) \in T^2 / (a_i \in A_\alpha^-) \wedge (a_i \in A_\omega^+), i \in \{1, 2\}$$

Cette relation peut être directement étendue sur l'ensemble des flux, F , de la façon suivante:

$$\forall R, \forall (f_1, f_2) \in F^2 / f_i = (a_1^{(i)}, \dots, a_n^{(i)}), f_1 | f_2 \Leftrightarrow \exists (t_\alpha, t_\omega) \in T^2 / (a_i^{(i)} \in A_\alpha^-) \wedge (a_n^{(i)} \in A_\omega^+), i \in \{1, 2\}$$

On remarquera que la relation de parallélisme ainsi définie est une relation d'équivalence. Ainsi, la relation de parallélisme permet de définir formellement la notion de faisceau comme il suit.

Définition 5.13: Un faisceau est un ensemble de flux $\phi = (f_1, \dots, f_m)$ tel que:

$$\forall (i, j) \in \{1 \dots n\}^2, f_i | f_j, \text{ ce qui sera noté } f_1 | \dots | f_m$$

Les transitions t_α et t_ω sont appelées respectivement transitions de début et de fin du faisceau ϕ . L'ensemble des faisceaux d'un RdPFT sera noté Φ .

On notera que les relations « ; » et « | » peuvent être étendues de façon immédiate sur l'ensemble $F \cup \Phi$.

Définition 5.14: Un faisceau $\phi = (f_1, \dots, f_m)$ de transition finale t_ω est dit *élémentaire* si et seulement si :

$$\left(\forall (f_i, f_j) \in \phi^2, i \neq j, f_i \cap f_j = \emptyset \right)$$

Autrement dit un faisceau élémentaire est un faisceau ne contenant pas d'autres faisceaux non réduits à une unité de synchronisation. On notera Φ^e l'ensemble des faisceaux élémentaires d'un RdPFT.

Définition 5.15: Un faisceau élémentaire $\phi = (f_1, \dots, f_m)$ est dit *équilibré* si et seulement si:

$$\forall f_i \in \phi, IVT_f(f_i) = (x_i, n, y_i)$$

Autrement dit, les flux composant un faisceau élémentaire équilibré ont une durée nominale identique. Les faisceaux équilibrés sont des structures de synchronisation fondamentales pour la spécification de schémas de synchronisation inter-flux continus. En effet, la nécessité de satisfaire la continuité des flux impose que les points de synchronisation inter-flux soient disposés périodiquement entre les flux avec une période égale à un multiple du plus petit multiple commun des durées nominales des unités de synchronisation composant ces flux .

Définition 5.16: On appelle forme canonique d'un faisceau élémentaire ϕ le faisceau élémentaire ϕ' vérifiant la propriété suivante:

$$\forall \phi = (f_1, \dots, f_n) \in \Phi^e / f_i = (a_1^{(i)}, \dots, a_n^{(i)}) = (f_1^{(i)}, a_n^{(i)}), \phi' = (f_1', \dots, f_n')$$

$$\text{avec } \forall i \in \{1, \dots, n\}, (f_i' = (b_1^{(i)}, b_2^{(i)})) \wedge (b_1^{(i)} = r_f(f_1^{(i)})) \wedge (b_2^{(i)} = a_n^{(i)})$$

On notera $C: \Phi^e \rightarrow \Phi^e$ la fonction associant à tout faisceau élémentaire ϕ sa forme canonique $C(\phi)$.

Propriété 5.2: Tout faisceau élémentaire peut être mis sous une forme canonique unique.

Démonstration: Nous allons fournir une procédure effective de mise sous forme canonique d'un faisceau élémentaire. Tout faisceau élémentaire peut être aisément mis sous forme canonique en appliquant l'algorithme de réduction suivant:

$$\forall \phi = (f_1, \dots, f_n) \in \Phi^e,$$

- $\forall f_i = (a_1^{(i)}, \dots, a_n^{(i)}) = (f_1^{(i)}, a_n^{(i)}) \in \phi / |f_i^{(i)}| \geq 1$ remplacer $f_i = (f_1^{(i)}, a_n^{(i)})$ par $f_i' = (r_f(f_1^{(i)}), a_n^{(i)})$. Autrement dit cette étape consiste à réduire tous les flux élémentaires du faisceau considéré.
- $\forall f_i \in \phi / |f_i| = 1$, remplacer $f_i = (a_n^{(i)})$ par $f_i' = (a_0^{(i)}, a_n^{(i)})$, avec $IVT(a_0^{(i)}) = (0, 0, 0)$

Ainsi cet algorithme transforme tout flux d'un faisceau élémentaire en un flux composé de deux unités de synchronisation. L'unité de synchronisation initiale de chaque flux ainsi transformé modélise de façon abstraite le comportement temporel des unités de synchronisation qui sont en amont des unités de synchronisation associées à la transition de fin du faisceau. Ainsi, comme nous le verrons ultérieurement, la forme canonique d'un faisceau (Figure 5.5-a et 5.5-b) permet de calculer l'intervalle de validité temporelle d'un faisceau ainsi que de vérifier aisément les propriétés temporelles d'un schéma de synchronisation inter-flux.

CQFD.

Tout comme la notion d'intervalle de validité temporelle d'une unité de synchronisation et d'un flux ont été définies précédemment, la notion d'intervalle de validité temporelle d'un faisceau peut être introduite. Cependant la définition de cette notion fait non seulement intervenir les intervalles de validité temporelle des flux constituant le faisceau considéré mais aussi la sémantique de synchronisation associée à la transition terminale du faisceau.

Propriété 5.3: La durée d'un faisceau élémentaire $\phi = (f_1, \dots, f_n)$ de transition finale t_ω peut prendre ses valeurs dans l'intervalle de validité temporelle défini par la fonction IVT_ϕ telle que:

$$IVT_\phi: \Phi^e \rightarrow \mathcal{Q}^+ \times \mathcal{Q}^+$$

$$\forall \phi = (f_1, \dots, f_n) \in \Phi^e / f_i = (a_1^{(i)}, \dots, a_n^{(i)}), IVT(a_k^{(i)}) = (x_k^{(i)}, n_k^{(i)}, y_k^{(i)}), IVT_f(f_i) = (X_i, N_i, Y_i)$$

soit $C(\phi)$ la forme canonique de ϕ , $C(\phi) = (g_1, \dots, g_n)$ avec $g_i = (b_1^{(i)}, b_2^{(i)})$.

$$\begin{aligned}
IVT(b_i^{(i)}) &= (X_i - x_n^{(i)}, N_i - n_n^{(i)}, Y_i - y_n^{(i)}) = (x_i, n_i, y_i) \text{ et } IVT(b_2^{(i)}) = IVT(a_n^{(i)}) \\
\text{si } \text{Syn}(t_\omega) &= \text{et} \text{ alors } IVT_\phi(\phi) = \left(\max_i(X_i), *, \max(\min(Y_i), \max(y_i)) \right) \\
\text{si } \text{Syn}(t_\omega) &= \text{et-faible} \text{ alors } IVT_\phi(\phi) = \left(\max_i(X_i), *, \max(Y_i) \right) \\
\text{si } \text{Syn}(t_\omega) &= \text{ou} \text{ alors } IVT_\phi(\phi) = \left(\max(\min(X_i), \max(x_i)), *, \max(Y_i) \right) \\
\text{si } \text{Syn}(t) &= \text{ou-fort} \Rightarrow IVT_\phi(\phi) = \left(\max(\min(X_i), \max(x_i)), *, \max(\min(Y_i), \max(y_i)) \right) \\
\text{si } \text{Syn}(t) &= \text{maître} \Rightarrow IVT_\phi(\phi) = \left(\max(X_m, \max(x_i)), *, \max(Y_m, \max(y_i)) \right) \\
\text{si } \text{Syn}(t) &= \text{ou-maître} \Rightarrow IVT_\phi(\phi) = \left(\max(\min(X_i), \max(x_i)), *, \max(X_m, \max(y_i)) \right) \\
\text{si } \text{Syn}(t) &= \text{et-maître} \Rightarrow IVT_\phi(\phi) = \left(\max(X_i), *, \max(Y_m, \max(y_i)) \right) \\
\text{si } \text{Syn}(t) &= \text{maître-faible} \Rightarrow IVT_\phi(\phi) = \left(\max(X_m, \max(x_i)), *, \max(Y_i) \right) \\
\text{si } \text{Syn}(t) &= \text{maître-fort} \Rightarrow IVT_\phi(\phi) = \left(\max(X_m, \max(x_i)), *, \max(\min(Y_i), X_m, \max(y_i)) \right)
\end{aligned}$$

Avec f_m le flux contenant l'arc maître associé à t_ω .

$\forall \phi \in \Phi^c$, $IVT_\phi(\phi)$ est appelé *intervalle de validité temporelle du faisceau* ϕ .

Démonstration: Cette propriété résulte immédiatement de la traduction des sémantiques de tir des RdPFT (chapitre 4, section 2) à l'aide des intervalles de validité temporelle des flux constituant le faisceau considéré. En considérant que l'origine des temps est la date de tir de la transition initiale du faisceau, une analyse temporelle de la forme canonique d'un faisceau ϕ nous permet d'établir immédiatement les inégalités suivantes:

$$\begin{aligned}
\forall i \in \{1, \dots, n\}, & (X_i \leq \tau_i^{\min} \leq y_i + x_n^{(i)}) \wedge (x_i + y_n^{(i)} \leq \tau_i^{\max} \leq Y_i) \wedge (\max_i(x_i) \leq \tau^{\text{min}} \leq \max_i(y_i)) \\
\Rightarrow & \left[\left(\min_i(X_i) \leq \min_i(\tau_i^{\min}) \leq \min_i(y_i + x_n^{(i)}) \right) \wedge \left(\max_i(X_i) \leq \max_i(\tau_i^{\min}) \leq \max_i(y_i + x_n^{(i)}) \right) \right] \\
& \left[\left(\min_i(x_i + y_n^{(i)}) \leq \min_i(\tau_i^{\max}) \leq \min_i(Y_i) \right) \wedge \left(\max_i(x_i + y_n^{(i)}) \leq \max_i(\tau_i^{\max}) \leq \max_i(Y_i) \right) \right]
\end{aligned}$$

Ainsi en considérant l'intervalle de tir associé à la transition finale du faisceau ϕ , tel qu'il est défini dans le chapitre 4 (Section 2) à partir des instants fondamentaux de synchronisation, on obtient directement l'intervalle de validité temporelle du faisceau en remplaçant les bornes inférieure et supérieure de ces intervalles exprimées à partir des instants fondamentaux par respectivement la valeur minimum et maximum susceptible d'être prise par cet instant de synchronisation. Ainsi, si nous prenons par exemple la définition, à l'aide des instants de synchronisation fondamentaux, de l'intervalle de tir associé au type « et-faible » nous avons $\tau^{\text{syn}} \in \left[\max_i(\tau_i^{\min}), \max_i(\tau_i^{\max}) \right]$, les inégalités précédentes nous permettent alors de déterminer que l'intervalle de

validité temporelle d'une transition de type « et-faible » est donné par $\left[\max_i(X_i), *, \max_i(Y_i) \right]$. Autrement dit les

bornes inférieure et supérieure de l'intervalle de validité temporelle d'un faisceau sont obtenues en associant une sémantique de tir à la transition finale t_0 et en déterminant les bornes supérieure et inférieure de l'intervalle de validité temporelle du faisceau considéré à partir, respectivement, d'une analyse temporelle optimiste (*i.e.* toutes les unités de synchronisation se terminent au plus tôt) et pessimiste (*i.e.* toutes les unités de synchronisation se terminent au plus tard). Par souci de simplification la durée nominale a été remplacée par le caractère « * » dans les formules précédentes. En effet, l'obtention de la durée nominale de ces intervalles de validité temporelle est obtenue en supposant que toutes les unités de synchronisation ont pour durée leur durée nominale. Le calcul de cette durée nominale ferait donc appel à un point de vue « exécution » permettant d'obtenir la date exacte d'un point de synchronisation à partir de la date de terminaison de chaque unité de synchronisation.

CQFD.

Propriété 5.4: Tous les flux d'un faisceau élémentaire ont même durée quelle que soit la sémantique de synchronisation associée à la transition de fin du faisceau considéré.

Démonstration:

$$\forall \phi = (f_1, \dots, f_n) \in \Phi^e / f_i = (a_1^{(i)}, \dots, a_n^{(i)})$$

avec $\tau_{i,k}^{deb}$ et $\tau_{i,k}^{syn}$ respectivement les dates de sensibilisation de la place associée à l'arc $a_i^{(k)}$ et de tir de la transition associée à l'arc $a_i^{(k)}$. Ainsi nous avons:

$$\forall \phi = (f_1, \dots, f_n) \in \Phi^e, \forall f_i = (a_1^{(i)}, \dots, a_n^{(i)}) \in \phi, d(f_i) = \sum_{j=1}^{n_i} d(a_j^{(i)}) = \sum_{j=1}^{n_i} (\tau_{j,i}^{syn} - \tau_{j,i}^{deb})$$

$$\text{or, } \forall i, \forall j \in (1, \dots, (n-1)_i), \tau_{j,i}^{syn} = \tau_{j+1,i}^{deb}$$

$$\Rightarrow \forall i, \sum_{j=1}^{n_i} (\tau_{j,i}^{syn} - \tau_{j,i}^{deb}) = \tau_{n_i,i}^{syn} - \tau_{1,i}^{deb}$$

$$\text{de plus, } \forall (k, l), (\tau_{1,k}^{deb} = \tau_{1,l}^{deb}) \wedge (\tau_{n_i,k}^{syn} = \tau_{n_i,l}^{syn})$$

$$\Rightarrow \forall (f_k, f_l) \in \phi^2, d(f_k) = d(f_l)$$

CQFD.

Cette propriété met en évidence le fait qu'un point de synchronisation inter-flux induit un recalage temporel des flux mis en jeu.

La durée d'un flux fait intervenir les instants de synchronisation et non les dates de terminaison des unités de synchronisation composant ce flux. Ainsi la propriété précédente met en évidence que tous les flux d'un faisceau élémentaire ont même durée. Cependant cette notion de durée fait abstraction des dates de terminaison des unités de synchronisation. Ainsi le fait que les flux d'un faisceau élémentaire ont même durée n'implique pas que toutes les unités de synchronisation terminales de ces flux se terminent à la même date. Cela signifie plutôt que le temps écoulé entre la date de début de chaque flux d'un faisceau et la poursuite des traitements, après avoir atteint le point de synchronisation inter-flux, est identique pour chaque flux. La prise en compte des décalages temporels pouvant être observés entre les dates de terminaison des unités de synchronisation impliquées dans un point de synchronisation inter-flux est réalisée à partir de la notion de dérive inter-flux. La *dérive inter-flux* entre deux flux appartenant à un faisceau élémentaire peut être définie, selon un point de vue exécution, comme la différence entre les dates de terminaison des unités de synchronisation terminales de ces deux flux.

Propriété 5.5: En l'absence de synchronisation inter-flux la dérive temporelle entre deux flux prend ses valeurs dans l'intervalle temporel donné par la fonction Δ définie comme il suit:

$$\begin{aligned} \Delta: F \times F &\rightarrow Q \times Q \\ \forall \phi \in \Phi^e, \forall (f_i, f_j) \in \phi^2 / IVT_f(f_k) = (x_k, n_k, y_k), k \in \{1,2\} \\ \Delta(f_i, f_j) &= (x_i - y_j, y_i - x_j) \end{aligned}$$

Démonstration:

En l'absence de synchronisation inter-flux nous avons:

$$\begin{aligned} (1) \left\{ \begin{array}{l} IVT_f(f_i) = (x_i, n_i, y_i) \Rightarrow x_i \leq d_f(f_i) \leq y_i \\ IVT_f(f_j) = (x_j, n_j, y_j) \Rightarrow x_j \leq d_f(f_j) \leq y_j \end{array} \right. \\ (1) \Rightarrow x_i - y_j \leq d_f(f_i) - d_f(f_j) \leq y_i - x_j \\ \Rightarrow \Delta(f_i, f_j) = (x_i - y_j, y_i - x_j) \end{aligned}$$

CQFD.

L'intervalle temporel ainsi défini détermine la plage de variation du décalage temporel susceptible d'intervenir entre les deux flux en l'absence de mécanisme de synchronisation. L'association d'une des 9 sémantiques de synchronisation à la transition finale, notée t_ω , du faisceau ϕ induit une resynchronisation et un recalage temporel des flux composant ϕ . Ce recalage temporel résulte d'une « accélération » ou d'un « ralentissement » de certains flux afin de satisfaire la sémantique de synchronisation associée à la transition finale du faisceau considéré. Ainsi, si l'on considère à titre d'exemple un faisceau élémentaire $\phi = (f_1, \dots, f_n) / \forall f_i \in \phi, IVT_f(f_i) = (x_i, n_i, y_i)$ ayant une transition finale t_ω de type « et-faible » (*i.e.* telle que $\text{Syn}(t_\omega) = \text{et-faible}$), alors un flux f_i subit un retard maximum lorsqu'il se termine à sa date de terminaison au plus tôt et lorsque tous les autres flux se terminent à leur date de terminaison au plus tard. Le retard maximum pouvant être subi par f_i est donc égal à $\max\left(0, \max_{k \neq i} (y_k) - x_i\right)$. En privilégiant le flux le plus en retard la sémantique de synchronisation de type « et-faible » n'est susceptible d'induire aucune interruption, assimilable à une accélération, du flux le plus en retard. Ainsi un flux f_i ne peut-il en aucun cas être « avancé » par une sémantique de synchronisation de type « et-faible ».

Définition 5.16: La durée d'un faisceau élémentaire est donnée par la fonction d_ϕ définie comme il suit:

$$\begin{aligned} d_\phi: \Phi^e &\rightarrow Q^+ \\ \forall \phi = (f_1, \dots, f_m) \in \Phi, d_\phi(\phi) &= d_f(f_1) = \dots = d_f(f_m) \end{aligned}$$

De plus, par définition, nous avons $\forall \phi \in \Phi^e, d_\phi(\phi) \in IVT_\phi(\phi)$

Propriété 5.6: $\forall \phi = (f_1, \dots, f_n) \in \Phi^e, \forall f_k \in \phi / IVT_f(f_k) = (x_k, n_k, y_k), \delta(f_k) = n_k - d_\phi(\phi)$

Démonstration:

Nous avons par définition $\forall f_k \in F, \delta(f_k) = n_k - d_f(f_k)$

or $\forall f_k \in \phi, d_f(f_k) = d_\phi(\phi)$

donc $\forall f_k \in \phi, \delta(f_k) = n_k - d_\phi(\phi)$

CQFD.

Les notions de ralentissement et d'accélération d'un flux par rapport à un faisceau, ainsi que celles de retard et d'avance d'un flux par rapport à un autre peuvent être formalisées de la façon suivante:

$\forall \phi \in \Phi, \forall (f_i, f_j) \in \phi / f_k = (a_i^{(k)}, \dots, a_n^{(k)}), k = i, j,$

$\tau_{n_i, k}^{fn} - d_\phi(\phi) < 0 \Leftrightarrow f_k$ est ralenti par le flux ϕ

$\tau_{n_i, k}^{fn} - d_\phi(\phi) > 0 \Leftrightarrow f_k$ est accéléré par le flux ϕ

$(\tau_{n_i, j}^{fn} - \tau_{n_i, j}^{fn}) < 0 \Leftrightarrow f_i$ est en avance sur f_j

$(\tau_{n_i, j}^{fn} - \tau_{n_i, j}^{fn}) > 0 \Leftrightarrow f_i$ est en retard sur f_j

Ainsi $\tau_{n_i, k}^{fn} - d_\phi(\phi)$ représente le *recalage temporel* induit sur le flux f_k par la sémantique de synchronisation associée à la transition finale t_ω du faisceau ϕ . Selon un point de vue exécution on qualifera d'unité de synchronisation la plus en avance (respectivement la plus en retard) celle se terminant au plus tôt (respectivement au plus tard). Selon un point de vue modélisation l'unité de synchronisation la plus en avance (respectivement la plus en retard) correspond à celle atteignant au plus tôt (respectivement au plus tard) la borne supérieure de son intervalle de validité temporelle. De plus on dira qu'une unité de synchronisation $a_n^{(k)}$, impliquée dans un point de synchronisation inter-flux, est retardée si $\tau_{n_i, k}^{fn} < \tau^{syn}$.

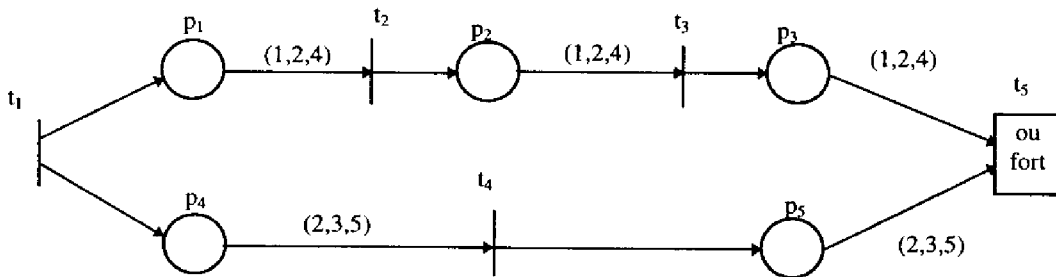


Figure 5.4: Un faisceau élémentaire et équilibré.

La notion de correction temporelle d'un faisceau résulte naturellement de celle de correction temporelle d'un flux.

Définition 5.17: Un faisceau est temporellement correct si et seulement si chacun des flux le constituant est temporellement correct.

Considérons à titre d'exemple le faisceau élémentaire, noté ϕ , illustré par la figure 5.4. Ce faisceau est composé de deux flux (i.e. $f_1 = (a_{1,1}, a_{1,2}, a_{1,3}) = ((p_1, t_2), (p_2, t_3), (p_3, t_5))$ et $f_2 = (a_{2,1}, a_{2,2}) = ((p_4, t_4), (p_5, t_5))$) de même durée nominale (i.e. 6 unités de temps), ce faisceau est donc équilibré. Le point de synchronisation inter-flux modélisé par la transition t_5 étant de type « ou-fort », l'instant de synchronisation inter-flux sera déterminé par l'unité de synchronisation la plus en avance (on suppose que l'implantation de cette sémantique de synchronisation est conforme au point de vue « exécution » spécifié dans le Chapitre 4 (section 4.3.3)). On supposera que les durées des unités de synchronisation composant ces flux sont les suivantes: $d(p_1, t_2)=1$, $d(p_2, t_3)=4$, $d(p_3, t_5)=1$, $d(p_4, t_4)=2$, $d(p_5, t_5)=3$. On remarquera que le flux f_2 est le plus en avance. Ainsi la terminaison de l'unité de synchronisation (p_5, t_5) détermine l'instant de synchronisation inter-flux et, par là même, la durée de l'unité de synchronisation (p_3, t_5) qui dans ce cas ne sera pas temporellement correcte. Les caractéristiques temporelles de ce faisceau sont résumées par le tableau 5.3.

Durée du faisceau ϕ	$d_\phi(\phi)=5$
Intervalle de validité temporelle du faisceau ϕ	$IVT_\phi(\phi)=(3, *, 10)$
Dérive du flux f_1 Recalage temporel de f_1	$\delta(f_1) = 6-6=0$ $6-5=1 \Rightarrow f_1$ est en accéléré de 1 unité de temps par t_5
Dérive du flux f_2 Recalage temporel de f_2	$\delta(f_2) = 5-6=-1$ $5-5=0$
Le faisceau n'est pas temporellement correct car le flux f_1 n'est pas temporellement correct	

Tableau 5.3: Caractérisation temporelle du faisceau illustré par la Figure 5.4.

5. Réduction et Vérification des RdPSFT

Nous allons montrer dans cette section que tout RdPSFT peut être réduit sous la forme d'un arc temporel. L'unité de synchronisation ainsi obtenue représente alors sous une forme abstraite le comportement temporel du RdPSFT objet de la réduction.

5.1 Réduction de Faisceaux Élémentaires

Nous allons montrer dans un premier temps comment tout faisceau élémentaire peut être réduit sous forme d'un arc temporel.

Propriété 5.7: Tout faisceau élémentaire peut être réduit sous la forme d'un faisceau composé d'un unique arc temporel à l'aide de la fonction de réduction suivante:

$$r_\phi: \Phi^c \rightarrow A^+$$

$$\forall \phi \in \Phi^c, r_\phi(\phi) = \phi' \left((\phi' = (f), f \in F) \wedge (f = (a), a \in A^+) \wedge (IVT(a) = IVT_\phi(\phi)) \right)$$

Démonstration: L'existence et l'unicité de la forme réduite est assurée par la définition de l'intervalle de validité temporelle d'un faisceau. Ainsi, tout faisceau élémentaire peut être transformé en un faisceau élémentaire unique

composé d'un seule unité de synchronisation représentant de façon abstraite le comportement temporel du faisceau (Figures 5.5-a et 5.5-b).

CQFD.

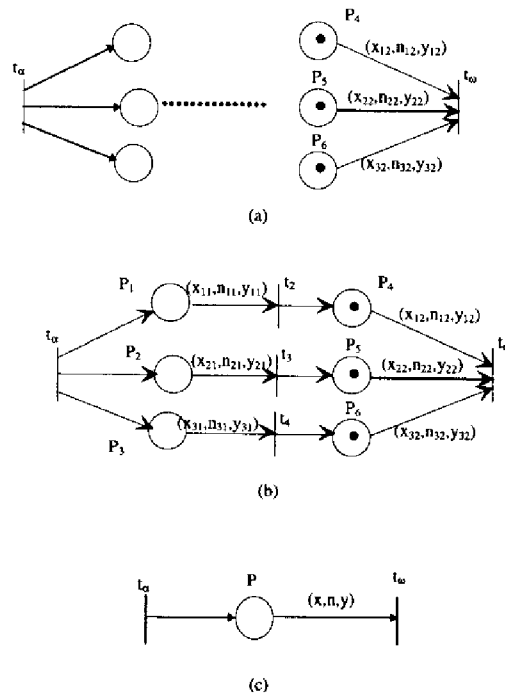


Figure 5.5: Réduction d'un faisceau. (a) Un faisceau élémentaire. (b) la forme canonique du faisceau. (c) le faisceau réduit en un faisceau composé d'un seul arc.

5.2 Réduction de Faisceaux et de RdPSFT

Définition 5.19: On notera par $R \triangleright^f R'$ (resp. $R \triangleright^\phi R'$) la transformation consistant à réduire le flux élémentaire f (resp. le faisceau élémentaire ϕ) dans le RdPSFT R . Autrement dit R' est obtenu en remplaçant dans R le flux f (resp. le faisceau ϕ) par l'arc a (resp. le faisceau ϕ') tel que $r_f(f) = a$ (resp. $r_\phi(\phi) = \phi'$). On notera \triangleright^\emptyset la transformation identité laissant inchangé tout RdPSFT (i.e. $\forall R, R \triangleright^\emptyset R$).

On remarquera que ces transformations induisent des règles de réécriture inverses de celles de construction descendante d'un RdPSFT consistant à pouvoir remplacer une place par l'un quelconque des 13 schémas de synchronisation fondamentaux. Ainsi ces réductions élémentaires transforment bien un RdPSFT en un nouveau RdPSFT.

Propriété 5.8: Tout faisceau d'un RdPSFT peut être réduit sous la forme d'un faisceau réduit à un unique arc temporel.

Démonstration: Nous allons montrer que tout faisceau peut être réduit sous la forme d'un unique faisceau élémentaire. La propriété résultera alors immédiatement du fait que tout faisceau élémentaire peut être réduit sous forme d'un unique arc temporel. Considérons la suite de réductions suivante appliquée à un faisceau ϕ :

$$\phi = \underbrace{\phi_1^{(1)} \triangleright \epsilon_1^{(1)} \dots \triangleright \epsilon_{n_1}^{(1)} \phi_{n_1}^{(1)}}_{\text{Etape 1}} \dots \underbrace{\phi_{n_{p-1}}^{(p-1)} \triangleright \epsilon_1^{(p)} \dots \triangleright \epsilon_{n_p}^{(p)} \phi_{n_p}^{(p)}}_{\text{Etape p}} = \phi'$$

Avec $\epsilon_i^{(j)}$ désignant un faisceau élémentaire appartenant au faisceau $\phi_i^{(j)}$.

Chaque itération de cette chaîne de réductions consiste à réduire tous les faisceaux élémentaires du faisceau considéré dans l'étape courante. Notons que pour chaque itération l'ordre des réductions des faisceaux élémentaires est quelconque. De plus chaque étape de cet algorithme réduit le cardinal de l'ensemble des places du faisceau examiné à l'étape courante. Le faisceau initial étant fini nous en déduisons donc que cet algorithme transforme, en un nombre fini de réductions élémentaires, tout faisceau en un faisceau ne contenant plus de faisceaux élémentaires. Ainsi, par définition, le faisceau résultant ϕ' est élémentaire, il peut donc être réduit en un seul arc temporel à l'aide de la réduction élémentaire $\phi' \triangleright a$.

Montrons de plus l'unicité de l'arc résultant de la réduction du faisceau ϕ . Chaque réduction élémentaire de la chaîne de réductions transforme un faisceau élémentaire en un faisceau ne possédant qu'une seule unité de synchronisation définie de façon unique par la fonction de réduction de faisceau élémentaire. Ainsi l'unicité des structures de synchronisation générées lors de chaque transformation garantit l'unicité de l'arc obtenu en extrémité de la chaîne des transformations.

CQFD.

L'expression de l'algorithme de réduction d'un RdPSFT sous la forme d'une suite de réductions élémentaires met en évidence que la complexité de l'algorithme de réduction est $O(n \times r)$, avec n le cardinal de l'ensemble des transitions t telles que $|t| > 1$ et r la complexité de l'algorithme de réduction d'un faisceau élémentaire. En considérant que l'algorithme de réduction d'un faisceau élémentaire consiste à calculer dans un premier temps les intervalles de validité temporelle des flux constituant le faisceau puis dans un second temps à appliquer les formules données par la propriété 5.1, on en déduit que la complexité de l'algorithme de réduction d'un faisceau élémentaire $\phi = (f_1, \dots, f_n)$ de transition finale t_ω est $O(k \times l + 2 \times (l+1))$ avec $k = \max_i |f_i|$ et $l = |t_\omega|$. Nous en déduisons donc que la complexité de l'algorithme de réduction d'un faisceau est d'ordre polynomial.

Corollaire 5.1: Tout faisceau peut être mis sous forme canonique.

Démonstration: Ce corollaire résulte immédiatement de la propriété précédente qui établit que tout faisceau peut être réduit sous la forme d'un faisceau élémentaire qui, en accord avec la propriété 5.6, peut être mis sous forme canonique.

CQFD.

Corollaire 5.2: Tout RdPSFT peut être réduit sous la forme d'un arc temporel.

Démonstration: Tout RdPSFT peut être mis sous la forme d'un RdPSFT comprenant une place d'entrée et une transition de sortie (éventuellement associées avec des intervalles de validité nuls). Ainsi tout RdPSFT mis sous cette forme possède un faisceau principal ϕ de plus haut niveau éventuellement réduit à un seul flux, et peut donc être décrit par la relation temporelle $(f_1; \phi; f_2)$ avec $f_1, f_2 \in F$ et $\phi \in \Phi$. Ainsi, d'après la propriété précédente, la réduction $R \xrightarrow{\phi} R'$ appliquée à R permet de changer le faisceau ϕ de R en faisceau composé d'un seul flux f_3 réduit à une seule unité de synchronisation. Le RdPSFT R' se réduit donc à un flux $f_4 = (f_1; f_3; f_2)$ pouvant être réduit sous forme d'un arc temporel, noté a, par la réduction $R' \xrightarrow{f_4} a$. Ainsi la suite de réductions $R \xrightarrow{\phi} R' \xrightarrow{f_4} a$ permet de réduire le RdPSFT R sous la forme d'un unique arc temporel. L'unicité de l'arc temporel ainsi obtenu est assurée par les Propriétés 5.5 et 5.8 établissant l'unicité de l'arc temporel obtenu respectivement suite à la réduction d'un flux et d'un faisceau.

CQFD.

Propriété 5.9: Pour toute structure de synchronisation s appartenant à l'ensemble $F \cup \Phi$, la relation \xrightarrow{s} est régulière à gauche.

Démonstration: Si R et R' sont des RdPSFT contenant le faisceau ou le flux s, tels que $R \xrightarrow{s} R''$ et $R' \xrightarrow{s} R''$ alors R et R' sont des RdPSFT contenant la structure de synchronisation s et égaux si l'on remplace s par l'arc a tel que $s \xrightarrow{a}$, donc $R=R'$.

CQFD.

Définition 5.20: On dira qu'un RdPSFT R se réduit en un RdPSFT R' s'il existe une suite $\left(\xrightarrow{s_1}, \dots, \xrightarrow{s_n} \right)$ de réductions telle que $\exists R_1, \dots, R_{n-1} \in RdPSFT / R \xrightarrow{s_1} R_1 \xrightarrow{s_2} \dots \xrightarrow{s_{n-1}} R_{n-1} \xrightarrow{s_n} R'$. On désignera par \triangleright cette relation sur l'ensemble des RdPSFT, et on notera $R \triangleright R'$ pour exprimer que le réseau R se réduit en R'.

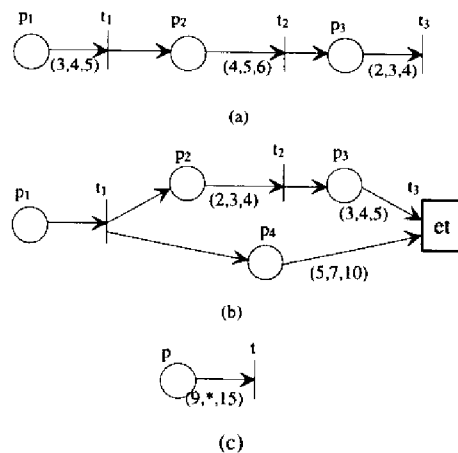


Figure 5.6: Réduction de deux RdPSFT (a) et (b), en un même arc temporel (c)

On remarquera que contrairement aux relations de réduction de structures de synchronisation notées \triangleright^s , la relation \triangleright n'est pas régulière à gauche. En effet deux réseaux différents peuvent être réduits en un même réseau. Ainsi, à titre d'exemple, les deux réseaux définis respectivement par les Figures 5.6-a et 5.6-b, bien que distincts, se réduisent en un même arc temporel (Figure 5.6 -c). Ainsi nous avons bien $(R_1 \triangleright R) \wedge (R_2 \triangleright R) \wedge (R_1 \neq R_2)$.

Propriété 5.10: La relation \triangleright est une relation d'ordre partiel.

Démonstration: La relation \triangleright n'est pas totale en effet, par exemple, le réseau de la figure 5.6-b ne peut pas être réduit en celui de la figure 5.6-a.

Nous allons montrer que cette relation est réflexive, antisymétrique et transitive.

- La relation \triangleright est réflexive; en effet:

$$\forall R \in RdPSFT, R \triangleright^{\emptyset} R \Rightarrow R \triangleright R$$

- La relation \triangleright est antisymétrique; en effet:

$$\begin{aligned} \forall (R, R') \in RdPSFT^2 / (R \triangleright R') \wedge (R' \triangleright R) \\ (R \triangleright R') \Rightarrow |P| \geq |P'| \\ (R' \triangleright R) \Rightarrow |P'| \geq |P| \\ (R \triangleright R') \wedge (R' \triangleright R) \Rightarrow |P| = |P'| \end{aligned}$$

Avec $|P|$ (resp. $|P'|$) désignant le cardinal de l'ensemble des places du RdPSFT R (resp. R'). Or la seule réduction conservant la cardinalité d'un RdPSFT est la réduction identité, donc $R=R'$.

- La relation \triangleright est transitive; en effet :

$$\begin{aligned} \forall (R, R', R'') \in RdPSFT^3 / (R \triangleright R') \wedge (R' \triangleright R'') \\ (R \triangleright R') \Rightarrow \exists \left(\begin{array}{c} s_1 \quad s_n \\ \triangleright, \dots, \triangleright \end{array} \right) / R \triangleright \dots \triangleright R' \\ (R' \triangleright R'') \Rightarrow \exists \left(\begin{array}{c} s'_1 \quad s'_m \\ \triangleright, \dots, \triangleright \end{array} \right) / R' \triangleright \dots \triangleright R'' \\ \text{donc } (R \triangleright R') \wedge (R' \triangleright R'') \Rightarrow \exists \left(\begin{array}{c} s_1 \quad s_n \quad s'_1 \quad s'_m \\ \triangleright, \dots, \triangleright, \triangleright, \dots, \triangleright \end{array} \right) / R \triangleright \dots \triangleright R' \triangleright \dots \triangleright R'' \\ \Rightarrow R \triangleright R'' \end{aligned}$$

CQFD.

Cette relation d'ordre permet de représenter sous une forme plus abstraite un schéma de synchronisation. Ainsi, si R et R' sont deux RdPSFT tels que $R \triangleright R'$ alors le schéma de synchronisation R' est une représentation abstraite plus « grossière » du comportement temporel que R . Autrement dit R' représente la spécification R avec une granularité de synchronisation moins fine.

Propriété 5.11: La relation d'ordre \triangleright sur l'ensemble des RdPSFT est bien fondée (*i.e.* il n'y a pas de suite infinie strictement décroissante d'éléments de l'ensemble des RdPSFT).

Cette propriété permet en particulier l'utilisation du raisonnement par induction dans l'ensemble ordonné $(RdPSFT, \triangleright)$.

Démonstration: Un ensemble ordonné $O=(RdPSFT, \triangleright)$ est bien fondé si et seulement si toute partie non vide E de O admet au moins un élément maximal.

Soit $E=\{R_1, \dots, R_n\}$ une partie non vide de l'ensemble des RdPSFT. Nous allons montrer que:

$$\exists R_k \in E / \forall R_i \in E, R_k \neq R_i \Rightarrow \neg(R_i \triangleright R_k)$$

Procédons "ab absurdo", c'est à dire supposons que:

$$\forall R_k \in E, \exists R_i \in E, (R_k \neq R_i) \wedge (R_i \triangleright R_k)$$

donc pour tout R_k dans E on peut construire une suite décroissante infinie $S=(R_i^{(k)})_{i \in \mathbb{N}}$ d'éléments de E telle que :

$\dots \triangleright R_n^{(k)} \triangleright \dots \triangleright R_1^{(k)} \triangleright R_0^{(k)} = R_k$. Etant donné que $\forall i, R_{i+1}^{(k)} \neq R_i^{(k)}$ on en déduit que pour tout i , il existe une suite

de réductions élémentaires non réduite à la réduction identité transformant $R_{i+1}^{(k)}$ en $R_i^{(k)}$. Ainsi la suite S est constituée d'éléments tous distincts. On aboutit donc à la conclusion suivante :

$$\forall E \subset O, E \neq \emptyset, \exists S \subset E / |S| > |E|$$

et qui est un prédicat toujours faux.

Nous en déduisons donc que l'ensemble ordonné $(RdPSFT, \triangleright)$ est bien fondé.

CQFD.

La fonction de réduction de RdPSFT permet de généraliser à un RdPSFT quelconque la notion d'intervalle de validité temporelle.

Définition 5.21: L'intervalle de validité temporelle d'un RdPSFT est défini par la fonction IVT_r , telle que:

$$\begin{aligned} IVT_r: \Phi &\rightarrow (\mathcal{Q}^+) \times (\mathcal{Q}^+) \times (\mathcal{Q}^+) \\ \forall R \in RdPSFT / R \triangleright a, IVT_r(R) &= IVT(a) \end{aligned}$$

Définition 5.22: On dira que deux RdPSFT R et R' sont temporellement équivalents, et on notera $R \approx R'$, si et seulement si $\exists R'' \in TSPN / (R \triangleright R'') \wedge (R' \triangleright R'')$.

Propriété 5.12: La relation \approx est une relation d'équivalence.

Démonstration:

- La relation \approx est réflexive; en effet:

$$\begin{aligned} \forall R \in RdPSFT, R \triangleright R \\ \Rightarrow R \approx R \end{aligned}$$

- La relation \approx est symétrique, en effet:

$$\begin{aligned} \forall (R, R') \in RdPSFT^2, R \approx R' \\ \Leftrightarrow \exists R'' / (R \triangleright R'') \wedge (R' \triangleright R'') \\ \Leftrightarrow \exists R'' / (R' \triangleright R'') \wedge (R \triangleright R'') \\ \Leftrightarrow R' \approx R \end{aligned}$$

- La relation \approx est transitive, en effet:

$$\begin{aligned} & \forall (R, R', R'') \in \text{RdPSFT}^3 / (R \approx R') \wedge (R' \approx R'') \\ & \left\{ \begin{array}{l} (R \approx R') \Rightarrow \exists X \in \text{RdPSFT} / (R \triangleright X) \wedge (R' \triangleright X) \\ (R' \approx R'') \Rightarrow \exists Y \in \text{RdPSFT} / (R' \triangleright Y) \wedge (R'' \triangleright Y) \end{array} \right. \\ \text{or } & \forall R \in \text{TSPN}, \exists^1 a \in A^+ / (R \triangleright a) \Rightarrow (\forall X, (R \triangleright X) \Rightarrow (X \triangleright a)) \\ & \text{ nous avons donc par transitivité de la relation } \triangleright \\ & (R \triangleright X) \wedge (X \triangleright a) \Rightarrow (R \triangleright a) \\ & \text{ de même nous avons } (R' \triangleright a) \\ & \text{ nous avons donc montré que:} \\ & \exists a \in A^+ / (R \triangleright a) \wedge (R'' \triangleright a) \\ & \text{ donc } R \approx R'' \end{aligned}$$

CQFD.

Définition 5.23: On notera $[R]_{\triangleright}$ la classe d'équivalence du réseau R pour la relation \triangleright . Un représentant canonique de cette classe d'équivalence est l'unique arc a tel que $R \triangleright a$. On notera que ce représentant canonique est un point fixe pour la relation de réduction.

La relation \approx définit une relation d'équivalence temporelle entre RdPSFT. Notons que $R \triangleright a \Rightarrow R \approx a$, on parlera ainsi de l'arc temporellement équivalent à un RdPSFT (nous avons montré que pour tout RdPSFT cet arc existe et est unique). Ainsi cette relation d'équivalence permet de représenter de façon abstraite un ensemble d'unités de synchronisation logiquement et temporellement corrélées.

5.3 Exemple

Considérons à titre d'exemple le RdPSFT introduit dans le chapitre 4 (section 6.2, figure 4.16). La réduction sous forme d'une unité de synchronisation de ce RdPSFT est résumée par la Figure 5.7. Cette figure met en évidence les étapes suivantes:

Le RdPSFT initial est tout d'abord transformé de façon à posséder une place d'entrée et une transition de sortie (Figure 5.7-a). Le RdPSFT résultant est alors composé d'une place d'entrée, d'un faisceau principal, noté ϕ , parenthésé par les transitions t_1 et t_{10} , et d'une transition de sortie.

L'application de l'algorithme de réduction au faisceau induit les étapes suivantes:

- dans une première étape les 7 faisceaux élémentaires composant le RdPSFT initial R sont réduits sous forme de faisceaux composés d'une seule unité de synchronisation (Figure 5.7-b). Ainsi le réseau initial R subit la

suite de réductions élémentaires suivante: $R \triangleright R_1 \triangleright \dots \triangleright R'$, avec ϕ_1, \dots, ϕ_7 les 7 faisceaux élémentaires composant R .

- la deuxième étape est consacrée à la réduction de l'unique faisceau élémentaire, noté ϕ' , de transition initiale t_1 et de transition finale t_{10} , composant le RdPSFT R' résultant de la première étape. Ce faisceau est à son tour réduit sous forme d'un faisceau composé d'une unique unité de synchronisation (P, t_{10}) en appliquant la

réduction élémentaire $R \triangleright f$. Le RdSFT résultant, noté f , est un flux élémentaire composé de trois unités de synchronisation (Figure 5.7-c).

- Le flux élémentaire résultant de l'étape précédente est alors réduit en une unité de synchronisation, notée « a », à l'aide de la réduction de flux $f \triangleright a$ (Figure 5.7-e).

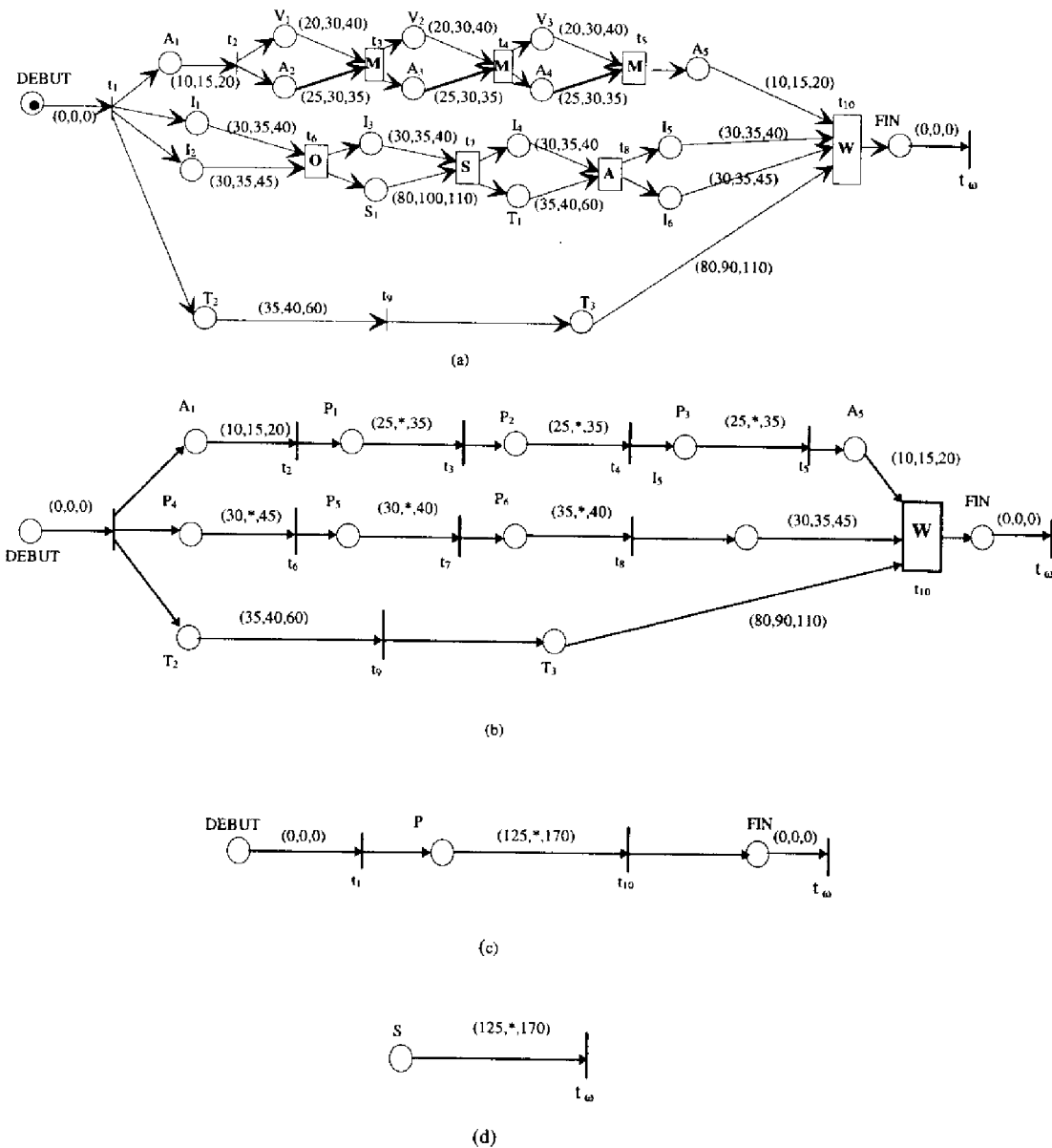


Figure 5.7: Réduction d'un RdPSFT. (a) le RdPSFT initial, (b) le RdPSFT obtenu après réduction des faisceaux élémentaires du RdPSFT initial, (c) le flux élémentaire obtenu après la deuxième étape de l'algorithme de réduction, (d) l'arc temporellement équivalent au RdPSFT initial.

5.4 Vérification Temporelle des Schémas de Synchronisation

Les sémantiques de synchronisation de type « best effort » introduites par le modèle RdPFT se justifient lorsque le spécifieur juge prohibitif le prix d'une vérification au regard de la faible criticité temporelle de sa spécification. En particulier l'algorithme de vérification proposé par [Wal83] est dans le cas général de complexité non-polynomiale. Dans un contexte « temps-réel mou » les paradoxes temporels introduits par ces sémantiques « best-effort » peuvent être jugés acceptables. Il n'en est plus de même pour les applications de type « temps-réel dur » pour lesquelles les contraintes de synchronisation doivent être renforcées afin d'éviter tout risque d'incohérence temporelle de (des) l'unité(s) de synchronisation que l'on cherche à privilégier. Ainsi, dans un système à forte criticité temporelle il est nécessaire de pouvoir garantir que les schémas de synchronisation inter-flux satisfont l'une des deux propriétés suivantes:

- Propriété 1: La (ou les) unité(s) de synchronisation que l'on cherche à privilégier voient toujours leurs contraintes temporelles satisfaites et ne peuvent être retardées.
- Propriété 2: La (ou les) unité(s) de synchronisation que l'on cherche à privilégier sont toujours temporellement correctes.

La deuxième propriété est moins restrictive que la première étant donné qu'elle permet d'accepter des schémas de synchronisation susceptibles de retarder des unités de synchronisation.

Syn(t)	Propriété 1	Propriété 2
et	$\exists \tau / \forall i \in I_{\omega}, \tau_i^{\min} = \tau_i^{\max} = \tau$	$\max_{i \in I_{\omega}}(\tau_i^{\min}) \leq \min_{i \in I_{\omega}}(\tau_i^{\max})$
et-faible	Toujours vérifiée	Toujours vérifiée
ou	$\min_{i \in I_{\omega}}(\tau_i^{\min}) \geq \tau^{\text{low}}$	$\min_{i \in I_{\omega}}(\tau_i^{\max}) \geq \tau^{\text{low}}$
ou-fort	$\min_{i \in I_{\omega}}(\tau_i^{\max}) \geq \tau^{\text{low}}$	$\min_{i \in I_{\omega}}(\tau_i^{\max}) \geq \tau^{\text{low}}$
maître	$\tau_m^{\min} \geq \tau^{\text{low}}$	$\tau_m^{\max} \geq \tau^{\text{low}}$
ou-maître	$\min_{i \in I_{\omega}}(\tau_i^{\min}) \geq \tau^{\text{low}}$	$\tau_m^{\max} \geq \tau^{\text{low}}$
et-maître	$\exists \tau / \forall i \in I_{\omega}, \tau_i^{\min} = \tau_m^{\max} = \tau$	$\tau_m^{\max} \geq \max_{i \in I_{\omega}}(\tau_i^{\min})$
maître-fort	$\tau_m^{\min} \geq \tau^{\text{low}}$	$\max_{i \in I_{\omega}}(\min(\tau_i^{\max}), \tau_m^{\min}) \geq \tau^{\text{low}}$
maître-faible	$\tau_m^{\min} \geq \tau^{\text{low}}$	$\tau_m^{\max} \geq \tau^{\text{low}}$

Tableau 5.4: Vérification des propriétés temporelles à l'aide des instants fondamentaux de synchronisation.

Ces deux propriétés peuvent être exprimées, en fonction du type de chaque transition t_{ω} telle que $|\bullet t_{\omega}| > 1$, à l'aide des instants fondamentaux de synchronisation selon le tableau 5.4.

La vérification des propriétés résumées par le tableau 5.4 peut être réalisée, pour tout schéma de synchronisation inter-flux d'un RdPSFT, en mettant sous forme canonique le faisceau associé à la transition finale du faisceau considéré. On peut alors réaliser une analyse pessimiste de la forme canonique à l'aide des intervalles de validité temporelle de ses unités de synchronisation. En effet chaque instant de synchronisation fondamental peut être directement traduit sous la forme d'une date relative à l'instant τ de tir de la transition initiale du faisceau considéré.

Soit $C(\phi)$ la forme canonique d'un faisceau ϕ d'un RdPSFT R . Nous avons vu dans la démonstration de la propriété 5.3 que la forme canonique $C(\phi)$ d'un faisceau ϕ permet d'obtenir des inégalités suivantes:

$$\begin{aligned} \forall \phi = (f_1, \dots, f_n) \in \Phi^* / \forall f_i = (a_i^{(i)}, \dots, a_n^{(i)}) \in \phi, IVT(a_k^{(i)}) = (x_k^{(i)}, n_k^{(i)}, y_k^{(i)}), IVT_j(f_i) = (X_i, N_i, Y_i) \\ \text{soit } C(\phi) \text{ la forme canonique de } \phi, C(\phi) = (g_1, \dots, g_n) \text{ avec } g_i = (b_1^{(i)}, b_2^{(i)}), \\ IVT(b_1^{(i)}) = (X_i - x_{n_i}^{(i)}, N_i - n_{n_i}^{(i)}, Y_i - y_{n_i}^{(i)}) = (x_i, n_i, y_i) \text{ et } IVT(b_2^{(i)}) = IVT(a_{n_i}^{(i)}) \\ \forall i \in \{1, \dots, n\}, (X_i \leq \tau_i^{\min} \leq y_i + x_{n_i}^{(i)}) \wedge (x_i + y_{n_i}^{(i)} \leq \tau_i^{\min} \leq Y_i) \wedge (\max_i(x_i) \leq \tau^{\max} \leq \max_i(y_i)) \\ \Rightarrow \left\{ \begin{array}{l} (\min_i(X_i) \leq \min_i(\tau_i^{\min}) \leq \min_i(y_i + x_{n_i}^{(i)})) \wedge (\max_i(X_i) \leq \max_i(\tau_i^{\min}) \leq \max_i(y_i + x_{n_i}^{(i)})) \\ (\min_i(x_i + y_{n_i}^{(i)}) \leq \min_i(\tau_i^{\max}) \leq \min_i(Y_i)) \wedge (\max_i(x_i + y_{n_i}^{(i)}) \leq \max_i(\tau_i^{\max}) \leq \max_i(Y_i)) \end{array} \right. \end{aligned}$$

Ainsi les propriétés de type 1 et 2 spécifiées dans le tableau précédent à partir de dates absolues, peuvent être vérifiées, conformément au Tableau 5.5, en utilisant les intervalles de validité temporelle des unités de synchronisation appartenant à la forme canonique.

Syn(t)	Propriété 1	Propriété 2
et	$(\forall i, j, (x_i = y_j = n_i = n_j) \wedge (x_{n_i}^{(i)} = x_{n_j}^{(j)} = n_{n_i}^{(i)} = n_{n_j}^{(j)}))$	$\max_j(y_i + x_{n_i}^{(i)}) \leq \min_j(x_i + y_{n_i}^{(i)})$
et-faible	Toujours vérifiée	Toujours vérifiée
ou	$\min_i(X_i) \geq \max_i(y_i)$	$\min_i(x_i + y_{n_i}^{(i)}) \geq \max_i(y_i)$
ou-fort	$\min_i(X_i) \geq \max_i(y_i)$	$\min_i(x_i + y_{n_i}^{(i)}) \geq \max_i(y_i)$
maître	$X_m \geq \max_i(y_i)$	$x_m + y_{n_m}^{(m)} \geq \max_{j \neq m}(y_{i,j})$
ou-maître	$\min_i(X_i) \geq \max_i(y_i)$	$x_m + y_{n_m}^{(m)} \geq \max_{j \neq m}(y_{i,j})$
et-maître	$(\forall i, j, (x_i = y_j = n_i = n_j) \wedge (x_{n_i}^{(i)} = x_{n_j}^{(j)} = n_{n_i}^{(i)} = n_{n_j}^{(j)}))$	$x_m + y_{n_m}^{(m)} \geq \max_i(y_i + x_{n_i}^{(i)})$
maître-fort	$X_m \geq \max_i(y_i)$	$\max(\min_j(x_i + y_{n_i}^{(i)}), X_m) \geq \max_i(y_i)$
maître-faible	$X_m \geq \max_i(y_i)$	$x_m + y_{n_m}^{(m)} \geq \max_{j \neq m}(y_{i,j})$

Tableau 5.4: Vérification de propriétés temporelles d'un faisceau à l'aide des intervalles de validité temporelle des unités de synchronisation composant la forme canonique du faisceau.

Nous en déduisons donc que le modèle RdPSFT permet de mettre en oeuvre avec une complexité polynomiale des techniques de vérification temporelle statique (*i.e.* sans générer de graphe d'accessibilité, autrement dit à partir d'une analyse structurelle du réseau).

6. Conclusion

Nous avons introduit dans ce chapitre une restriction du modèle RdPFT. Cette restriction recouvre tous les RdPFT construits en utilisant exclusivement de façon récursive les structures temporelles de Allen. Le modèle Réseau de Petri Structuré à flux temporel ainsi défini possède les propriétés suivantes:

- il permet de spécifier des schémas de synchronisation de complexité arbitraire à partir d'une méthode structurante et minimaliste induisant de bonnes propriétés structurelles pour le réseau de Petri résultant.
- Il est particulièrement bien adapté à la spécification de scénarios de synchronisation multimédia. En effet, il offre une traduction formelle des entités fondamentales mises en jeu dans un scénario multimédia (*i.e.* unités de synchronisation, flux, faisceaux.....). De plus ce modèle induit une méthode de spécification orientée flux de traitements qui est conforme au paradigme de la « régie de production ».
- des opérations de réduction peuvent être appliquées sur un RdPSFT. Ces opérations permettent de faire varier de façon cohérente la granularité de synchronisation associée à une spécification. En particulier, cela permet d'adapter une spécification aux ressources disponibles pour supporter l'application multimédia.
- Afin de mettre en évidence les problèmes potentiels de désynchronisation dans les schémas de synchronisation inter-flux, des techniques de vérification statique peuvent être appliquées à l'aide d'algorithmes de complexité polynomiale. L'existence et le « faible coût » de ces techniques de vérification temporelle font des RdPSFT un modèle bien adapté à la spécification d'application « temps-réel critique ».

Cependant le modèle RdPSFT n'en possède pas moins un certain nombre de limitations. En particulier ce modèle ne permet pas de spécifier des schémas de synchronisation non-déterministes. Il apparaît donc que ce modèle est bien adapté à la spécification des schémas de synchronisation multimédia statiques, c'est à dire se déroulant de façon immuable. Ce modèle ne permet pas la spécification de scénarios de synchronisation mettant en jeu des événements asynchrones issus de l'environnement et susceptibles de faire évoluer dynamiquement le comportement d'un système. Nous allons montrer dans le chapitre suivant comment le modèle RdPFT permet de modéliser des événements asynchrones dans un environnement temporellement contraint.

Chapitre 6

Les Réseaux de Petri Hiérarchisés à Flux Temporels

1. Introduction

Nous avons étudié dans les chapitres précédents la contribution des RdPFT pour la modélisation de scénarios de synchronisation multimédia. Ces scénarios « déroulent » des présentations immuables pour un utilisateur considéré comme un simple spectateur d'une représentation multimédia. La présentation d'information multimédia de façon intégrée et dynamique est une composante importante de tout système d'information, l'importance prise par l'audiovisuel dans notre société en témoignant. Cependant les limites de l'approche consistant à considérer l'utilisateur comme un sujet passif sont clairement identifiées, à savoir:

- la multiplicité des sources d'information, inhérente aux systèmes multimédia distribués, rend nécessaire de fournir à l'utilisateur des moyens d'accès interactif à l'information. Dans un tel environnement multimédia distribué l'utilisateur doit être considéré comme un acteur interagissant avec le système pour accéder à l'information. Le système multimédia apparaît alors comme un système réactif soumis aux sollicitations de l'utilisateur. De plus l'utilisateur doit pouvoir substituer aux contraintes temporelles intrinsèques à un scénario multimédia ses propres contraintes de temps. Autrement dit un utilisateur doit pouvoir contrôler temporellement le déroulement d'un scénario multimédia (on rejoint ici les primitives de contrôle temporel d'accès à un flux multimédia évoquées dans le Chapitre2).
- le structuralisme a mis en évidence que la connaissance s'acquiert, de façon privilégiée, à partir de processus d'apprentissage où l'individu construit « en marchant » son propre savoir. Ainsi, de façon à faciliter tout processus d'apprentissage, un système d'information multimédia doit permettre à l'individu d'être acteur de son propre savoir en lui offrant des mécanismes interactifs d'exploration de la connaissance. Le succès spectaculaire du World-Wide-Web témoigne d'un réel besoin des utilisateurs en matière de système interactif d'accès au savoir, ou plus généralement à l'information.

Ainsi un nouvel acteur principal émerge de la scène du multimédia; il s'agit de l'environnement qui est susceptible de modifier dynamiquement à partir d'événements asynchrones le comportement d'une application

multimédia. L'intégration des contraintes de synchronisation multimédia et de celles de l'environnement débouche sur une extension de la synchronisation multimédia que nous appellerons synchronisation hypermédia. Etant donné le caractère fondamental de l'environnement dans une application hypermédia, il est nécessaire d'intégrer dès le départ celui-ci dans toute tentative de modélisation des systèmes hypermédiés. Ainsi l'objet de ce chapitre est de montrer la contribution du modèle RdPFT pour la modélisation des systèmes hypermédiés [SDLS-S96].

2. Organisation du Chapitre

La première partie de ce chapitre est consacrée à une brève introduction aux notions de système et de synchronisation hypermédiés. Un panorama des techniques proposées pour la modélisation des systèmes hypermédiés est ensuite exposé. Nous montrons en particulier la contribution du formalisme des réseaux de Petri dans le domaine de la modélisation des systèmes hypermédiés. A partir des limitations des modèles existants et des contributions apportées par les Réseaux de Petri nous montrons alors comment le modèle RdPFT peut être étendu afin de permettre une modélisation aisée et complète des structures fondamentales d'un système hypermédia.

3. Introduction à l'Hypermédia

Face à l'explosion des connaissances, il est nécessaire de développer de nouvelles méthodes d'accès au savoir qui ne reposent plus sur un accès purement linéaire à l'information, mais qui permettent à partir de nouvelles structures organisationnelles et opérationnelles de documents, un accès rapide et naturel à l'information. Par accès naturel nous entendons un accès mettant en œuvre, lors d'un processus de recherche ou de présentation d'information, des structures opérationnelles les plus proches possible de schémas humains de pensée [Lév93]. Ainsi, idéalement, l'utilisateur et le concepteur ne doivent être contraints ni par les méthodes d'accès au document ni par la construction de celui-ci.

Ces nouvelles méthodes doivent permettre de maîtriser la complexité structurelle de vastes champs de connaissances. En particulier, la diversité des besoins et des profils d'utilisateurs impose de pouvoir aborder et agrémenter le document selon plusieurs points de vue.

Depuis le début de l'écriture, de telles méthodes ont été développées de différentes façons plus ou moins performantes et empiriques. On peut citer en particulier les thesaurus ou index encyclopédiques qui se proposent à travers des tables de références croisées d'indiquer les relations (plus ou moins étroites) existant entre divers champs encyclopédiques. Cependant de telles méthodes ne répondent que très partiellement aux critères précités quand elles sont confrontées à de gros volumes d'informations.

Parallèlement à l'évolution du champ des connaissances, les systèmes informatiques ont prouvé leur capacité à traiter de façon performante de gros volumes d'informations. Il paraît donc naturel d'exploiter le potentiel de traitement de l'information des systèmes informatiques afin de développer des environnements satisfaisant les deux critères précédents. De tels environnements répondent au nom d'hypertexte ou d'hypermédia.

3.1 Définition de la Notion de Système Hypermédia

Ted Nelson, un des pionniers de l'hypertexte le définit comme étant "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display...of a nonlinear text...which cannot be printed on a conventional page" [Nel67]. La définition du terme hypermédia peut être donnée, dans une première approche, en remplaçant dans la définition précédente le terme texte par multimédia. L'hypermédia apparaît à ce titre comme une généralisation de l'hypertexte. L'hypermédia est donc une technique de construction et de parcours d'un document multimédia permettant de rompre avec une structure de document purement séquentielle ou linéaire. Un document hypermédia apparaît comme un ensemble d'objets multimédia, ou *nœuds*, reliés par des *liens* sémantiques dont la gestion est prise en charge par le système hypermédia. L'utilisateur, suite à des activations de liens, peut ainsi « naviguer » parmi cet ensemble de nœuds multimédias. Une différence fondamentale doit être soulignée entre hypertexte et hypermédia. En effet, un document hypertexte ne contient aucune donnée dynamique; à ce titre toute activation d'un lien implique la présentation totale du nœud textuel qui lui est associé. Par contre, dans un système hypermédia deux niveaux de synchronisation se superposent, celui mis en œuvre par l'utilisateur et celui associé aux scénarios de synchronisation multimédia. Ainsi, dans le cadre d'un système hypermédia la présentation d'un nœud peut être modifiée ou interrompue par l'utilisateur.

Bien qu'il existe de nombreuses réalisations, industrielles ou émanant de laboratoires de recherche, de systèmes hypermédiés, il apparaît que la notion de système hypertexte vient juste d'être clarifiée par le modèle de référence Dexter [HS94] et que la notion de système hypermédia n'a pas été encore définie précisément. Ainsi on peut paraphraser le constat que fit Conklin en 1987 pour l'hypertexte : «hypermédia is at the beginning of something big, something like the invention of the wheel, but something that still has rough edges » [Con87].

3.2 Les Problèmes Soulevés par l'Hypermédia

Les systèmes hypermédiés en tant que support à la communication et à la pensée [Con87] ont pour objectif ultime la suppression des syndromes de surcharge et de défaut d'information provoqués par le volume et la diversité toujours croissants d'informations dans notre société [WB94]. Malheureusement ce but ambitieux ne peut être atteint trivialement à l'aide d'une simple mise en oeuvre des concepts de base des systèmes hypermédiés. En particulier, les systèmes hypermédiés introduisent deux artefacts allant, de façon paradoxale, à l'encontre des objectifs fondamentaux qu'ils poursuivent, à savoir:

- La *désorientation*, représentant la perte du sens de l'orientation que peut éprouver l'utilisateur lorsqu'il parcourt les méandres du labyrinthe d'information d'un document hypermédia. En effet, l'utilisateur doit à tout instant savoir où il se trouve dans le réseau d'informations et comment aller à une autre place dont il connaît ou soupçonne l'existence. Le problème de désorientation tient alors au fait qu'un document hypermédia, à la différence d'un document à structure purement linéaire, offre en chacun de ses lieux plusieurs degrés de liberté, ou dimensions, pour parcourir le document (un document à structure linéaire n'offrant que deux degrés de liberté, à savoir la recherche avant et la recherche arrière). Cette caractéristique induit une explosion combinatoire des chemins explorables à partir d'un lieu donné et rend, par là même, plus complexe l'appréhension de la structure globale du document par l'utilisateur.

- *La surcharge de connaissances.* Par ce terme on entend l'effort de réflexion que doit réaliser l'utilisateur pour effectuer un parcours pertinent et efficace du réseau informationnel face auquel il se trouve confronté. En effet, l'utilisateur doit à chaque nœud faire un choix parmi l'ensemble des chemins s'offrant à lui. Il doit mémoriser les chemins qu'il a empruntés et ceux qu'il sera pertinent de prendre ultérieurement; de plus il doit s'impliquer intellectuellement de façon à réfléchir sur la pertinence de suivre un lien. Cette problématique est encore plus aiguë quand l'utilisateur est amené à parcourir plusieurs liens et chemins en parallèle.

Deux catégories de solutions non exclusives ont été proposées pour résoudre ces problèmes :

- d'une part, porter principalement son effort sur l'ergonomie de l'interface utilisateur et l'utilisation de points de repères judicieusement choisis,
- d'autre part, superposer à un réseau hypermédia une sémantique de parcours. Cette sémantique de parcours, en filtrant et réduisant l'espace des choix offerts à l'utilisateur, sert de guide dans le parcours du document hypermédia.

En associant sémantique de parcours et structure du document, la deuxième catégorie de solution est la plus à même d'assurer l'ouverture, la portabilité, la robustesse, et plus généralement la qualité des systèmes hypermédiés. Cette constatation soulève le problème plus général de la spécification de la sémantique des systèmes hypermédiés.

De plus la complexité des systèmes hypermédiés est accrue par le fait qu'ils doivent gérer deux catégories de contraintes de synchronisation, à savoir:

- les contraintes de synchronisation temporelle qui doivent être respectées pour garantir une bonne restitution des scénarios de synchronisation multimédia. Ces scénarios de synchronisation sont susceptibles de mettre en oeuvre des schémas de synchronisation intra et inter-flux multimédia de complexité arbitraire.
- les contraintes de synchronisation logiques qui définissent les liens sémantiques entre les nœuds multimédiés, les événements présidant à l'activation de ces liens, et plus généralement la sémantique de parcours du réseau hypermédia.

Ainsi la maîtrise des systèmes hypermédiés passe de façon primordiale par celle des contraintes de synchronisation logique et temporelle intrinsèques à ces systèmes.

Consciente des enjeux de l'hypermédia la communauté informatique internationale a oeuvré pour la définition de structures de documents hypermédiés « ouverts ». Ainsi les normes MHEG [MHEG93] et Hytime [NKN91] définissent des structures de documents hypermédiés portables et accessibles de façon distribuée. Cependant ces normes ne sont pas en mesure à elles seules de résoudre la complexité des systèmes hypermédiés. Au contraire par leur caractère informel ou semi-formel, elles peuvent être source de complexité et d'erreurs.

Par contre, les techniques de spécification formelle peuvent apporter d'importantes contributions dans le domaine encore mal défini de l'hypermédia. En effet, les modèles formels devraient permettre de donner une sémantique

formelle aux entités fondamentales composant un système hypermédia. De plus, le caractère critique pour la sémantique applicative des schémas de synchronisation logique et temporelle que l'on peut rencontrer dans un système hypermédia, ainsi que leur complexité arbitraire, légitiment l'utilisation de techniques de spécification formelles permettant une spécification précise, rigoureuse et vérifiable des contraintes de synchronisation devant être satisfaites par le système. La section suivante est donc consacrée à une évaluation des principaux modèles qui ont été proposés jusqu'alors pour la spécification des contraintes de synchronisation dans les systèmes hypermédiés.

4. Panorama des Modèles Existants

4.1 Le Modèle de Référence « Hypertexte » Dexter

Le modèle Dexter fournit une terminologie standardisée ainsi qu'un modèle formel (exprimé en notation Z) pour les principaux concepts constitutifs d'un système hypertexte. Le modèle de référence Dexter [HS94] peut à ce titre être considéré comme un modèle de référence pour la conception de systèmes hypertextes. Conformément à la Figure 6.1, ce modèle de référence distingue trois niveaux dans un système hypertexte.

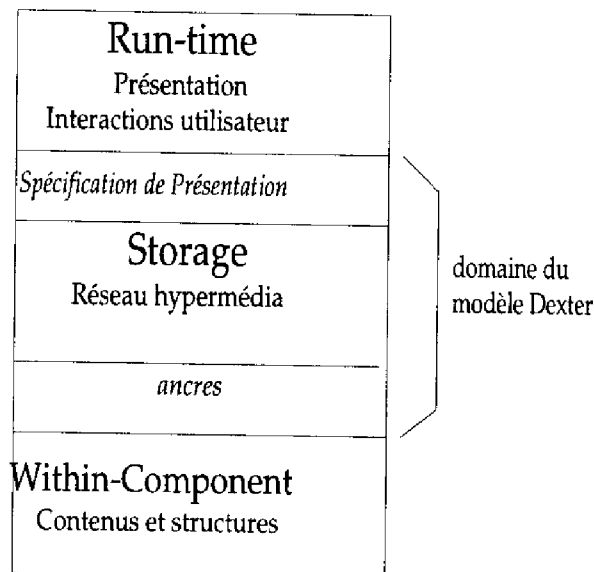


Figure 6.1: Le Modèle de référence hypertexte Dexter

En partant du plus haut niveau de la hiérarchie ainsi définie, ces trois niveaux sont les suivants:

- le niveau « *runtime* » : réalisant la présentation des entités hypermédiés ainsi que la gestion des interactions utilisateur. Ce niveau fortement dépendant de l'environnement utilisateur n'est traité par le modèle Dexter qu'au travers de la notion de « spécifications de présentation » permettant d'assurer l'interface entre le niveau « runtime » et le niveau sous-jacent par l'intermédiaire d'attributs de présentation associés aux « composants » hypertextuels (voir ci dessous la définition de composant).

- le niveau « *storage* »: constituant le point de focalisation du modèle Dexter. Ce niveau met en oeuvre une « base de données » de composants. Le concept de « composant » s'avère être la notion de base du modèle Dexter: il représente de façon unifiée les entités constitutives d'un réseau d'informations hypermédia, à savoir:
 - les « *composants atomiques* »: représentant des entités hypertextuelles élémentaires dont la structure interne n'est pas du domaine du modèle Dexter. Il en est ainsi par exemple d'un media de représentation de type JPEG ou MPEG.
 - les « *composants composites* »: permettant de définir des structures hiérarchisées acycliques à partir de composants atomiques et/ou composites. On peut déceler à l'état « embryonnaire » derrière la notion de noeud composite celle de scénario multimédia.
 - les « *composants liens* » permettant de mettre en oeuvre des relations n-aires entre des parties de composants atomiques ou composites.
- le niveau « *within component* » concerné par le contenu et la structure des composants. De façon à garantir l'indépendance du « niveau storage » par rapport à la diversité potentiellement infinie des formats de codage gérés par le niveau « within-component », les points d'attache d'un lien dans un composant atomique ou composite sont spécifiés à l'aide d'entités appelées « *ancres* ». Les « ancres » réalisent ainsi l'interface entre la structure logique du document hypertexte et la structure physique des composants atomiques et composites.

Si nous considérons qu'un système hypermédia est une extension naturelle d'un système hypertexte dans lequel les composants atomiques et composites correspondent respectivement à des données monomédias ou multimédias élémentaires et à des scénarios de présentation multimédias, il apparaît alors que Dexter pose les bases pour un modèle conceptuel plus général destiné aux systèmes hypermédias. Cependant la principale limitation du modèle de référence Dexter réside dans le fait qu'il ne permet pas de décrire le comportement dynamique d'un système hypermédia. Or, nous avons vu que les contraintes de synchronisation issues de l'environnement ou intrinsèques aux scénarios multimédias participent à la définition de la sémantique applicative, et à ce titre doivent être décrites par un modèle hypermédia. En particulier un modèle pour la synchronisation hypermédia doit permettre une spécification complète des contraintes temporelles internes à un système hypermédia [HBvR94]. Cependant, la description de ces aspects dynamiques est déportée par le modèle Dexter dans le « *Within layer* » ou bien à l'interface entre le « *storage layer* » et le « *runtime layer* » sous la forme de spécifications de présentations primitives. Ainsi, le modèle Dexter ne permet de décrire un système hypermédia que comme un « réseau » de noeuds multimédias encapsulant de façon opaque leurs contraintes de synchronisation. Le modèle Dexter ne permet donc pas de modéliser explicitement les contraintes temporelles d'un système hypermédia; par là même Dexter n'offre pas ce qu'on est en droit de demander à un modèle formel pour les systèmes hypermédias, à savoir la capacité à modéliser, simuler, analyser et vérifier le comportement dynamique des systèmes hypermédias.

4.2 Le Modèle Hypermédia d'Amsterdam

Considérant que le modèle Dexter pose des bases incomplètes pour la modélisation des systèmes hypermédias le Modèle Hypermédia d'Amsterdam (MHA) a apporté des extensions au modèle Dexter (principalement sous la forme d'extensions apportées aux spécifications de présentation) permettant la modélisation des contraintes de synchronisation dans un système hypermédia [HBvR94]. En particulier, le MHA souligne la nécessité de pouvoir modéliser les contraintes temporelles des composants atomiques, les contraintes de synchronisation internes aux composites, ainsi que les contraintes de synchronisation entre composants (i.e. composants atomiques ou composites).

Ainsi le MHA met en évidence la nécessité de modéliser dans un système hypermédia deux types de synchronisation:

- la synchronisation logique permettant la navigation basée sur des activations de liens, et
- la synchronisation temporelle permettant de mettre en oeuvre des scénarios de synchronisation de complexité arbitraire entre, ou à l'intérieur de composants composites ou atomiques.

Les principales différences entre ces deux types de synchronisation sont résumées par le Tableau 6.1.

<i>Synchronisation Temporelle</i>	<i>Synchronisation Logique</i>
date des points de synchronisation connues a priori (avec une certaine variabilité temporelle dans les systèmes faiblement synchrones)	date des événements de synchronisation non connues a priori
basée sur des événements internes aux composants atomiques ou composites (par exemple la fin de la présentation d'un composant atomique)	basée sur des événements externes (par exemple une interaction utilisateur)
relations temporelles entre des composants atomiques et composites (i.e. synchronisation inter et intra-flux)	relations logiques (ou liens) entre composants atomiques et composites ou à l'intérieur d'un composite, permettant la navigation entre ces composants.
Décrit le comportement dynamique intrinsèque aux composants atomiques ou composites.	Entraîne l'interruption et/ou l'activation du comportement dynamique des composants atomiques ou composites.

Table 6.1 : Principales différences entre synchronisation logique et temporelle

Le MHA introduit plusieurs extensions au modèle Dexter qui permettent une modélisation très complète des contraintes de synchronisations logique et temporelle. Bien que dotées d'un bon pouvoir d'expression ces extensions telles que les attributs temporels des composants atomiques, les attributs de synchronisation des composants composites, les arcs temporels et les contextes de liens, rendent difficile la perception globale des contraintes de synchronisation internes à un système hypermédia. En effet, le MHA induit la dissémination de ces contraintes de synchronisation qui se trouvent de plus exprimées sous diverses formes dans une spécification. Ainsi, le MHA souffre du manque d'une approche permettant une modélisation unifiée des contraintes de synchronisations logique et temporelle dans un système hypermédia. De plus le MHA ne propose pas une sémantique formelle permettant de vérifier et analyser les propriétés logiques et temporelles d'un système hypermédia avant de le concevoir.

4.3 Les Apports des Réseaux de Petri pour la Modélisation Hypermédia

4.3.1 Considérations Générales

Les RdP s'avèrent être un modèle formel particulièrement bien adapté pour la description d'architectures de documents hypermédias. En effet un réseau de Petri peut être considéré selon deux points de vue:

- en tant que graphe biparties; à ce titre il permet de décrire les liens existant entre les diverses unités d'information du document, autrement dit la structure hypertexte du document.
- en tant qu'automate étendu; à ce titre il permet de décrire, à partir de la notion d'états et des règles d'évolution entre états la sémantique de parcours du document hypermédia.

Cette dualité permet de décrire formellement des architectures de documents hypermédias en leur associant une sémantique de parcours et de synchronisation permettant de résoudre les problèmes précédemment évoqués, de désorientation de l'utilisateur et de surcharge d'information. De tels problèmes découlent de structures de documents hypermédias élaborées en s'appuyant uniquement sur les liens existant entre les diverses unités d'information du document.

La Figure 6.2 donne un exemple de graphe orienté qui spécifie les liens existant entre les nœuds multimédias d'un document hypermédia. Ce graphe fortement connexe indique qu'à partir de l'un quelconque des trois nœuds du réseau hypermédia on peut directement accéder à n'importe quel autre nœud.

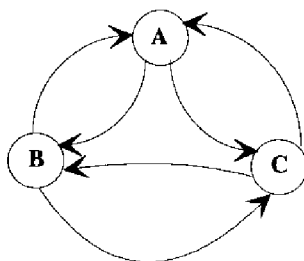


Figure 6.2 : Un réseau hypermédia

Ce graphe recouvre toutes les sémantiques de parcours admissibles du document hypermédia. En effet ce graphe peut donner lieu à plusieurs sémantiques de parcours décrites par autant d'automates déterministes. A ce titre la Figure 6.3 donne deux automates décrivant deux sémantiques de parcours différentes mais compatibles avec le graphe décrit par la Figure 6.2. La Figure 6.3-b décrit un automate dont le graphe sous-jacent est isomorphe au graphe décrivant le réseau hypermédia. L'automate ainsi généré induit la sémantique de parcours la moins contrainte, relativement au réseau hypermédia, et offre à l'utilisateur le maximum de degrés de liberté dans le parcours du réseau hypermédia. Cette traduction brute du réseau hypermédia sous la forme d'un automate isomorphe n'aide donc en rien pour la résolution des problèmes de désorientation et de surcharge d'information.

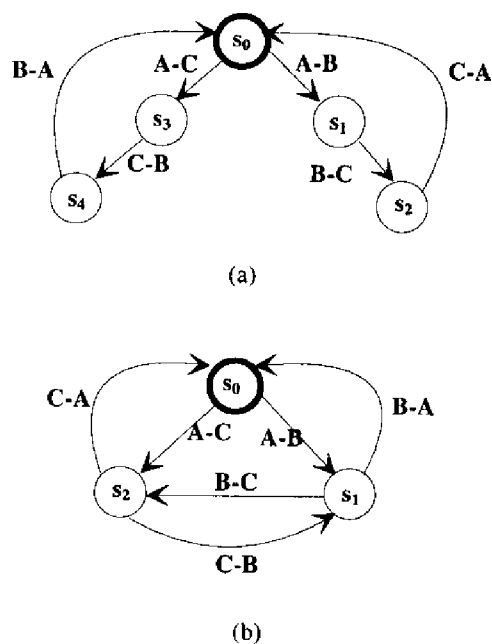


Figure 6.3 : Deux sémantiques de parcours compatibles avec le réseau de la Figure 6.2

Par contre la Figure 6.3-a impose une sémantique de parcours sur le réseau hypermédia. En effet cet automate n'offre qu'une transition lorsqu'il est dans un état différent de l'état initial (l'état s_0). Ainsi cet automate guide l'utilisateur, à partir de l'état initial, à travers les deux chemins A,B,C,A et A,C,B,A. En réduisant le nombre de choix à chaque nœud et en assurant un retour systématique à l'état initial, cet automate contribue grandement à résoudre les problèmes de désorientation et de surcharge d'information. La multiplicité des interprétations d'un réseau d'information hypermédia induit des sémantiques de parcours fortement dépendantes de l'implémentation et des « browsers ». Ce problème de portabilité de la sémantique de parcours est perturbant pour l'utilisateur qui ne peut s'appuyer sur une sémantique de parcours bien définie.

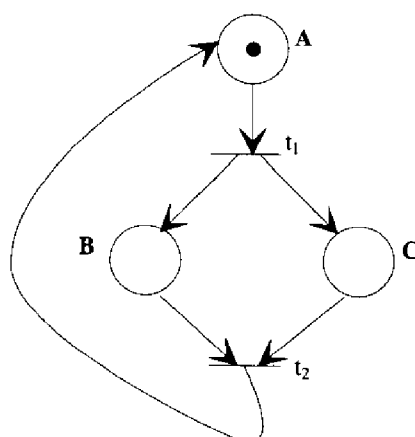


Figure 6.4 : Sémantique de parcours définie à partir d'un réseau de Petri et n'offrant qu'une seule possibilité de cheminement.

Une interprétation sous forme d'automate n'est pas en mesure d'induire une sémantique de parcours optimale sur le document hypermédia. En effet, se pose toujours le problème de faciliter la progression de l'utilisateur à travers plusieurs chemins issus d'un même nœud source et parcourus successivement. Une spécification sous forme d'automate ne permet pas de spécifier les chemins corrélés et devant, ou pouvant, faire l'objet d'une investigation simultanée ou parallèle. Autrement dit, une spécification sous forme d'automate ne permet pas de guider l'utilisateur dans des parcours simultanés à travers le réseau hypermédia. A cette fin il est nécessaire de disposer d'un formalisme permettant de spécifier des activités concurrentes et synchronisées. Ainsi le formalisme des réseaux de Petri peut apporter une contribution à la résolution des problèmes de désorientation et de surcharge d'information. A titre d'exemple la Figure 6.4 donne une traduction sous forme de Réseau de Petri du réseau hypermédia de la Figure 6.2. Cette spécification, tout en respectant le graphe hypermédia, réduit l'espace des choix laissés à l'utilisateur en lui proposant de traverser parallèlement les deux chemins induits par l'automate de la Figure 6.3-a. Ainsi, pour tout état du réseau de Petri, une seule possibilité de parcours s'offre à l'utilisateur.

4.3.2 Le Modèle Trellis

Les réseaux de Petri Temporels [Mer74] ont été utilisés dans [SF90], sous une forme interprétée appelée Trellis, afin de modéliser la « sémantique de parcours » d'un hypertexte. Ce travail a montré que l'utilisation des réseaux de Petri pour la modélisation des hypertextes permet de réduire la « surcharge d'information » et la « désorientation » des utilisateurs. De plus, le pouvoir d'analyse des réseaux de Petri temporels permet la vérification des propriétés logiques et temporelles des hypertextes. La principale limitation du modèle Trellis réside dans le fait que ce modèle s'attache uniquement à la description des contraintes de synchronisation logique internes à un document hypertextuel. Dans Trellis un lien est modélisé sous la forme d'une transition associée à un intervalle temporel, de plus les composants atomiques ou composites sont modélisés sous la forme de places élémentaires.

Dans le modèle Trellis, conformément aux règles de tir des réseaux de Petri temporels, un lien (i.e. une transition) d'intervalle associé $[a,b]$ n'est activable (i.e. tirable) à la date τ^{syn} que, si tous les composants « origines » du lien (i.e. le préfixe de la transition) ont débuté leur présentation (i.e. sont marquées, soit τ^{ou} le plus petit des instants réalisant cette condition), et si $\tau^{ou} + a \leq \tau^{syn} \leq \tau^{ou} + b$. Ce modèle, initialement développé pour la spécification de documents hypertextuels, ne prend pas en compte le caractère éventuellement dynamique des composants « origines » du lien et les traite comme des données statiques présentées de façon instantanée à l'utilisateur. Autrement dit, le modèle Trellis permet de spécifier uniquement les contraintes temporelles associées aux liens et fait abstraction des contraintes de synchronisation des composants atomiques et composites.

Le pouvoir d'expression du modèle Trellis apparaît donc limité vis à vis de la description de la synchronisation temporelle dans les documents hypermédiés.

4.4 Pourquoi un Nouveau Modèle ?

Nous avons introduit précédemment deux des modèles les plus évolués et éprouvés pour la spécification de systèmes hypermédiés. Le MHA illustre une approche semi-formelle ayant un bon pouvoir d'expression mais non

minimaliste (en terme de nombre de concepts utilisés). Inversement, le modèle Trellis représente une approche formelle minimaliste mais incomplète pour la modélisation hypermédia.

	MHA	Trellis	OCPN	RdPFT	Esterel	Extensions temporelles à LOTOS
<i>formalisme graphique</i>	oui	oui	oui	oui	non	non
<i>modèle formel</i>	non	oui	oui	oui	oui	oui
<i>pouvoir d'expression pour la synchronisation temporelle</i>	les synchronisation intra-media et continues ne sont pas considérées	les contraintes de synchronisation temporelles des composants atomiques et composites ne sont pas considérées	bon	bon	bon	bon
<i>pouvoir d'expression pour la synchronisation logique</i>	bon	limité à l'hypertexte	pas l'objet de ce modèle	pas l'objet de ce modèle	bon	bon
<i>pouvoir de modélisation pour la synchronisation temporelle</i>	modéré, contraintes de synchronisation trop dispersées	faible, contraintes temporelles pour les liens seulement	bon, paradigme de composition de la « régie de production numérique »	bon, paradigme de composition de la « régie de production numérique »	faible, formalisme des langages synchrones	faible, formalisme des algèbres de processus
<i>Pouvoir de modélisation pour la synchronisation logique</i>	bon, mise en correspondance immédiate avec Dexter	bon pour les composants atomiques et les liens, mais faible pour les composites	pas l'objet de ce modèle	pas l'objet de ce modèle	faible, formalisme des langages synchrones	faible, formalisme des algèbres de processus
<i>Hypothèse de synchronisme faible</i>	oui	oui	non, utilisation de durées nominales	oui	non, hypothèse de synchronisme	oui
<i>Constructeurs fondamentaux utilisés pour la modélisation multimédia et hypermédia¹</i>	attributs temporels, attributs de synchronisation, arcs temporels	transitions temporisées	places temporisées et transitions	arcs temporisés et transitions typées	signaux, opérateurs de synchronisation	portes, opérateurs de synchronisation et temporels

Table 6.2: Evaluation de quelques modèles pour la synchronisation multimédia et hypermédia

Plus généralement une étude détaillée des principales techniques ayant été proposées pour la modélisation multimédia et hypermédia permet de dégager les considérations suivantes:

- un formalisme graphique tel que MHA ou Trellis facilite la modélisation et la compréhension des scénarios de synchronisation hypermédia.
- La complexité croissante des systèmes hypermédiés rend nécessaire l'utilisation de méthodes formelles pour vérifier et valider leurs propriétés logiques et temporelles. Les réseaux de Petri Temporels, les algèbres de processus telles que les extensions temporelles à LOTOS [CdO94], ou les langages synchrones tels

¹ Ceci donne une idée du nombre de constructeurs que le spécifieur doit maîtriser afin de modéliser des schémas de synchronisation multimédia ou hypermédia.

qu'ESTEREL [SHH92], permettent de modéliser formellement et d'analyser les contraintes de synchronisation dans un système hypermédia.

- Les systèmes distribués hypermédiés sont des systèmes faiblement synchrones. Ainsi, un modèle pour la spécification des contraintes de synchronisation dans un système hypermédia doit permettre d'exprimer l'indéterminisme temporel des traitements. De plus cette possibilité offre plus de souplesse en matière de gestion des paramètres de Qualité de Service.
- Une technique de modélisation des systèmes hypermédiés doit permettre d'exprimer facilement et complètement à partir d'un formalisme unificateur les contraintes de synchronisation logique et temporelle. On rejoint par là la nécessité de disposer d'un modèle ayant de bons pouvoirs de modélisation et d'expression.

Le tableau 6.2 résume comment les principales méthodes de spécification satisfont ces critères. Ce tableau met en évidence qu'il n'existe pas de modèle satisfaisant l'ensemble des critères souhaitables. Plus précisément ce Tableau montre qu'à ce jour il n'existe pas de formalisme graphique permettant, à l'aide d'une approche unifiée, une modélisation aisée et complète des contraintes de synchronisation dans un système hypermédia. Cependant on peut remarquer que le modèle RdPFT satisfait tous les critères recherchés dans un modèle pour la synchronisation multimédia ; de plus, nous n'avons pas confronté pour l'instant ce modèle à la modélisation des systèmes hypermédiés. Ainsi, en considérant les RdPFT sous l'éclairage des concept fondateurs définis par le modèle de référence hypertexte Dexter, nous allons montrer dans les sections suivantes comment le modèle RdPFT peut être étendu afin de devenir un modèle hypermédia satisfaisant l'ensemble des critères précédemment énoncés.

5. Modélisation des Composants Atomiques

La présentation des composants atomiques (considérés par le modèle Dexter comme des entités primitives à l'intérieur du niveau « storage ») tels que des flux audio, vidéo, des images, etc., est assujettie à des contraintes temporelles intrinsèques aux médias ou dépendantes de l'application. De plus, les systèmes hypermédiés doivent être considérés, dans le cas général, comme des systèmes distribués faiblement synchrones dans lesquels les composants atomiques sont répartis sur plusieurs serveurs et combinés lors de l'exécution. Nous avons vu dans le chapitre 2 que dans un contexte faiblement synchrone il est irréaliste d'associer aux traitements des instants et des durées exacts. Il est plus cohérent d'utiliser des intervalles temporels pour spécifier les contraintes temporelles. De plus l'utilisation d'intervalles de validité temporelle induit une tolérance de synchronisation qui permet de concevoir des documents hypermédiés pouvant être adaptés plus facilement aux ressources disponibles. Une telle approche dans la modélisation des contraintes de synchronisation temporelles est prônée par le modèle RdPFT. En effet nous avons vu dans le Chapitre 4 que les RdPFT permettent d'associer des Intervalles de Validité Temporelle (i.e. IVT) à des traitements modélisés sous forme d'arcs temporels. Ainsi les RdPFT permettent de modéliser un composant atomique sous la forme d'une unité de synchronisation (Figure 6.5). Les composants atomiques peuvent être distingués dans un RdPFT en associant un type de ressource atomique (i.e. audio, vidéo, image....) aux places relatives à de telles unités de synchronisation.

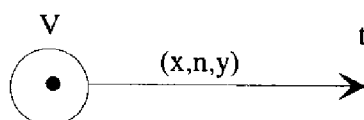


Figure 6.5: Modélisation d'un composant atomique.

6. Modélisation des Composants Composites

Selon le modèle de référence Dexter, les composants composites sont construits récursivement à partir de composants atomiques et composites. La modélisation des composants composites nécessite donc des possibilités de modélisation hiérarchisée. Nous avons vu dans le chapitre 5 que les réseaux de Petri de base offrent des possibilités de modélisation hiérarchique à l'aide de places représentant de façon abstraite un sous-réseau structurellement équivalent (i.e. le sous-réseau peut être substitué à la place). Ainsi, en étendant le formalisme des RdPFT, les composants composites peuvent être modélisés par l'intermédiaire d'un nouveau type de places, appelées places composites, associées de façon sous-jacente à un RdPFT décrivant un scénario de synchronisation multimédia.

Etant donné le rôle fondamental joué par le temps dans les systèmes hypermédias, les places composites doivent offrir non seulement des possibilités d'abstraction structurelle mais également temporelle. Nous avons vu dans le chapitre 5 que les RdPSFT sont particulièrement bien adaptés à la modélisation de scénarios multimédias (i.e. de composants composites). De plus ce chapitre a mis en évidence que tout RdPSFT peut être réduit sous la forme d'un arc temporellement équivalent, résumant de façon abstraite le comportement temporel du RdPSFT. D'où l'idée de modéliser les composants composites à l'aide de places abstraites associées de façon sous-jacente à un RdPSFT.

Plus précisément, cette approche hiérarchique possède la sémantique formelle suivante:

Pour définir une place composite C associée à un RdPSFT R , on utilise l'algorithme suivant:

- Soit R un RdPSFT (Figure 6.6.a). S'il ne possède pas de place d'entrée, lui associer un arc d'entrée, noté « e », tel que $IVT(e)=(0,0,0)$. Ajouter dans tous les cas à R un arc de sortie, noté s , tel que $IVT(s)=(0,0,0)$ (Figure 6.6-b). Soit R' le RdPSFT ainsi obtenu. Remarquons que R' est structurellement équivalent à un arc et que tout RdPSFT peut se mettre sous cette forme.
- Calculer, à partir de l'algorithme de réduction proposé dans le chapitre 5, l'arc « a » tel que $R' \triangleright a$, c'est à dire l'arc temporellement équivalent à R' . Soit $IVT(a)=(x,n,y)$ l'intervalle de validité temporelle de l'arc « a » résultant (Figure 6.6-c).
- La place composite C est alors telle que son sous réseau sous-jacent, R'' , est obtenu en supprimant la transition et l'arc de sortie de R' (Figure 6.6-d). De plus, tout arc a_i issu de C doit satisfaire la relation $IVT(a_i)=IVT(a)$; le type composite et une référence vers R'' sont associés à C .

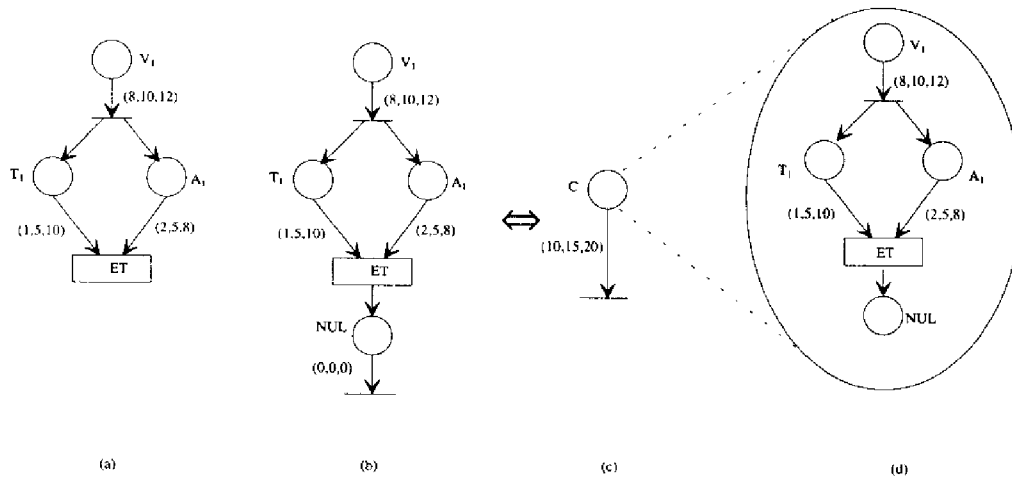


Figure 6.6 : Equivalence structurelle et temporelle entre un sous réseau et une place composite

A l'inverse toute place composite peut être remplacée par son sous-réseau en appliquant l'algorithme suivant:

- Donner la valeur $(0,0,0)$ aux intervalles de validité temporelle des arcs issus de la place multimédia.
- Remplacer la place composite par son sous-réseau.

La modélisation des composants composites sous forme de RdPSFT permet d'appliquer les techniques de vérification temporelle introduites dans le Chapitre 5 à la partie la plus temporellement critique d'un système hypermédia: les scénarios de synchronisation multimédia.

7. Modélisation des Liens

Le modèle de référence Dexter définit les liens hypertextuels comme des relations n-aires entre des portions de composants atomiques ou composites. Les liens définissent à la fois des relations sémantiques entre composants multimédias et des capacités de navigation entre ces composants offertes à l'utilisateur. Cette fonction duale des liens induit la notion de « sémantique de parcours » d'un document hypermédia qui distingue au sein du réseau hypertextuels un scénario de cheminement privilégié. Nous allons montrer dans cette section que le formalisme RdPFT permet non seulement de modéliser les systèmes hypertextuels traditionnels, mais qu'il apporte également de nombreuses et importantes contributions pour la modélisation et la conception des systèmes hypermédiés.

7.1 Liens Temporels

Les liens hypertextuels traditionnels définissent une relation sémantique entre des composants sources et destination, de plus ces liens sont associés à une condition logique d'activation dont la satisfaction assure le « franchissement » du lien. Le franchissement d'un lien entraîne la cessation de la présentation des composants sources au profit de l'activation de la présentation des liens destinations. Cette sémantique des liens hypertextuels trouve une correspondance immédiate dans le modèle RdPFT sous la forme suivante :

- un arc temporel, $a=(L,t)$, associé à un nouveau type de place appelé lien, tel que $IVT(a)=(0,*,+\infty)$ et $MA(t)=a$, modélise un lien et la condition d'activation² associée au lien,
- la transition t de type « maître » explicite la relation sémantique et la possibilité de franchissement existant entre les places de son préfixe modélisant les composants sources du lien et les places de son suffixe modélisant les composants destination du lien.

Notons que la date d'activation d'un lien est fonction de conditions portant sur le contexte et l'environnement applicatif. A ce titre la date nominale d'activation d'un lien ne peut être connue a priori, aussi cette date est remplacée par le caractère « * » dans l'IVT d'un arc modélisant un lien. Le caractère « * » dans un IVT signifie n'importe quelle valeur entre les bornes de cet intervalle. Ainsi, par exemple, le RdPFT de la Figure 6.7 modélise un lien (i.e. l'arc (L,t_1)), les deux composants « sources » du lien (i.e. les arcs (C_1,t_1) et (C_2,t_1)), et les deux composants « destination » du lien (i.e. les arcs (C_3,t_2) et (C_4,t_3)). Le formalisme des RdPFT permet donc de modéliser aisément les *liens multi-sources et multi-destinations*.

Une telle modélisation des liens permet de plus d'introduire la notion de *lien temporel*. Par définition, les liens temporels [SF90] sont des liens qui doivent être activés à l'intérieur d'un intervalle temporel. En effet, la sémantique de synchronisation de type « maître » spécifie que la transition associée doit être tirée dans l'Intervalle Absolu de Validité Temporelle (IAVT) de l'arc maître (i.e. du lien). Si nous prenons par exemple le RdPFT de la Figure 6.7, et si l'arc « maître » (L,t) est sensibilisé à la date τ , alors la transition t_1 doit être tirée dans l'intervalle temporel $[\tau+a, \tau+b]$. Si la condition d'activation du lien n'a pas été satisfaite au plus tard à l'instant $\tau+b$, alors le lien est automatiquement activé (i.e. la transition t_1 est tirée) à l'instant $\tau+b$ (si la transition est toujours sensibilisée à cette date). Il est important de noter que la condition d'activation d'un lien temporel est toujours considérée comme fausse avant l'instant $\tau+a$ et comme satisfaite à l'instant $\tau+b$. Ainsi les liens temporels permettent de mettre en oeuvre des systèmes hypermédiés actifs, c'est à dire susceptibles d'activer automatiquement des liens en fonction de conditions temporelles. Les liens temporels induisent ainsi une sémantique de parcours sur un document hypermédia, et sont donc susceptibles d'assurer une fonction de « guidage » de l'utilisateur dans son parcours du document hypermédia.

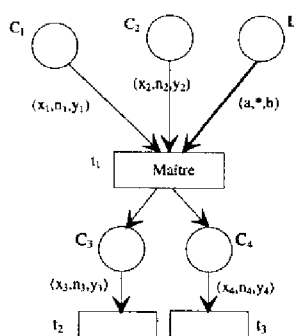


Figure 6.7 : Modélisation d'un lien multi-sources et multi-destinations.

² Les conditions logiques d'activation des liens peuvent être modélisées sous la forme de prédicats associés aux arcs.

Ainsi le modèle RdPFT permet d'adopter un *formalisme unifié* permettant la modélisation des composants (i.e. liens, composants atomiques ou composites) sous l'unique forme d'arcs temporels composés de places typées (i.e. places de types atomique, composite ou liens).

7.2 Liens Dynamiques

La sémantique traditionnelle de synchronisation logique dans les systèmes hypertextes induit des schémas de synchronisation exclusivement assujettis à la satisfaction de conditions logiques et/ou temporelles portant sur les liens. Cependant, en introduisant des composants dynamiques l'hypermédia rend nécessaire l'utilisation de *liens dynamiques* c'est à dire des liens activables en fonction de l'état temporel des composants auxquels ils sont attachés [Lie94].

Nous avons vu que le modèle RdPFT permet de modéliser facilement et complètement à l'aide d'un formalisme unificateur, à la fois les contraintes de synchronisation logique que l'on rencontre dans les hypertextes, et les contraintes de synchronisation temporelle des systèmes multimédias. Cependant, le type de synchronisation logique que nous avons modélisé précédemment à l'aide de la synchronisation de type « maître » est directement hérité de l'hypertexte, et considère à ce titre les composants atomiques et composites comme des données statiques. Autrement dit ce type de synchronisation logique fait abstraction du comportement dynamique des composants atomiques et composites. L'intégration des contraintes temporelles des composants dans la sémantique de synchronisation logique permettrait de bénéficier de schémas de synchronisation logique non seulement « conduits » par les liens mais aussi par les contraintes temporelles des composants. Une telle possibilité permettrait, par exemple, de spécifier des schémas de synchronisation tels que: « un lien peut être activé si et seulement si tous ses composants « sources » ont atteint la borne inférieure de leurs intervalles de validité temporelle ». Un tel schéma de synchronisation pourrait être utile si, par exemple, l'auteur de l'hypermédia voulait obliger l'utilisateur à « voir » un minimum des composants « sources » d'un lien avant de pouvoir activer ce lien. Il apparaît ainsi que l'intégration du comportement dynamique des composants atomiques ou composites dans les schémas de synchronisation logique est susceptible d'apporter une contribution importante aux systèmes hypermédias.

Or cette intégration du comportement dynamique des composants dans la définition des contraintes de synchronisation logique peut être réalisée aisément à l'aide du modèle RdPFT. En effet, nous avons vu dans le Chapitre 4 que le modèle RdPFT propose une déclinaison de la synchronisation de type « maître » permettant de combiner de façon complète les contraintes temporelles d'un lien, modélisé sous forme d'un arc temporel « maître », avec celles des autres arcs appartenant au même préfixe de transition. On notera de plus que n'importe laquelle des 9 sémantiques de synchronisation introduites par le modèle RdPFT peut être utilisée de façon à modéliser un schéma de synchronisation logique mêlant les contraintes de synchronisation logiques/temporelles d'un lien et les contraintes de synchronisation temporelles des composants « sources » de ce lien. On peut ainsi considérer que ces 9 sémantiques de synchronisation utilisées initialement dans un cadre multimédia fournissent également une sémantique formelle de la notion de *synchronisation hypermédia*.

Nous allons plus particulièrement étudier l'apport des 5 sémantiques de synchronisation dérivées du type « maître » (i.e. maître, et-maître, ou-maître, maître-fort, maître-faible) dans un contexte de synchronisation logique. Cette étude sera faite en analysant les effets respectifs de ces 5 types de synchronisation sur le simple³ schéma de synchronisation illustré par la Figure 6.10. Dans cette figure, $\text{Syn}(t)$ représente un type de synchronisation pris dans l'ensemble {maître, et-maître, ou-maître, maître-fort, maître-faible} et ayant l'arc (L,t) comme arc maître. L'arc (C,t) représente le composant (atomique ou composite) source du lien (L,t) .

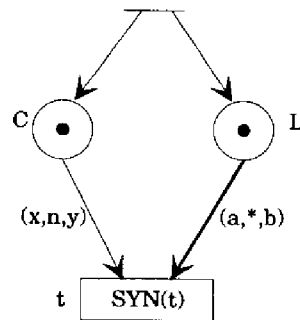


Figure 6.10: Un schéma de synchronisation hypermédia.

7.2.1 Taxonomie de la Synchronisation dans les Systèmes Hypermédiias

7.2.1.1 La Synchronisation de Type « Maître »

Ce type de synchronisation est issu de l'hypertexte et correspond à celui qui est communément utilisé dans les documents hypermédiias. Un tel type de synchronisation logique est uniquement conduit par les contraintes temporelles du lien (l'arc maître (L,t) dans la Figure 6.10). Relativement à l'exemple de la Figure 6.10, la sémantique de synchronisation logique, si la transition t est de type « maître » avec (L,t) comme arc « maître », est la suivante:

- Le lien peut être activé si la transition est sensibilisée et si le lien est dans son IVT (i.e. $(a,*,b)$).
- Si le lien n'a pas été activé avant la borne maximum de son IVT et si la transition est sensibilisée (i.e. le système hypermédia a lancé la présentation du composant C) alors le lien est automatiquement activé.

Ainsi l'activation du lien L dans l'intervalle relatif $[a,b]$ peut interrompre à tout instant la présentation du composant C, et ceci indépendamment des contraintes temporelles de la présentation multimédia.

7.2.1.2 La Synchronisation de Type « Maître-fort »

Les liens de type « maître-fort » tout comme les liens de type « maître » doivent être activés à l'intérieur de leur IVT. Mais à la différence des liens « maîtres » leur condition d'activation est satisfaite si un composant

³ Si nous considérons que l'instant de tir de la transition initiale de ce faisceau est $\tau=0$, alors pour chaque arc de ce faisceau élémentaire nous avons $\text{IVT}=\text{IAVT}=\text{IDVT}$.

« source » est terminé ou bien si la condition logique associée au lien est satisfaite. Ainsi, à la différence de la synchronisation logique de type « maître », la synchronisation de type « maître-fort » tient compte des contraintes de synchronisation des composants « sources » du lien . En particulier, la sémantique de synchronisation associée à la transition t de type « maître-fort » du RdPFT de la Figure 6.10 est la suivante:

- le lien peut être activé au plus tôt à la borne minimum de son IVT, c'est à dire « a ».
- le lien doit être activé au plus tard à la date $\max(a, \min(b, y))$

Ainsi si nous prenons par exemple $(x, y) = (10, 20)$ et $(a, b) = (15, 30)$, la transition est tirable dans l'intervalle $[15, 20]$; en particulier la transition t sera automatiquement franchie à la date relative 20 même si la condition logique associée au lien n'est pas satisfaite. Cette condition de franchissement étant alors induite par le composant C et non par le lien L . Ainsi, avec ce type de synchronisation, la première présentation multimédia se terminant ou atteignant la borne maximum de son intervalle de validité temporelle est susceptible de déclencher la synchronisation logique à l'intérieur de l'intervalle de validité temporelle du lien associé.

7.2.1.3 La Synchronisation de Type « Ou-Maître »

L'idée intuitive derrière la synchronisation de type « ou-maître » est d'activer un lien dès que la présentation d'au moins un des composants « sources » est terminée ou dès que la condition logique associée au lien est satisfaite dans son IVT. Ce type de synchronisation induit une forme d'indéterminisme de synchronisation borné supérieurement par la borne supérieure de l'IVT associé au lien. Ainsi, la sémantique de synchronisation logique associée à la transition t de type « ou-maître » du RdPFT de la Figure 6.10 est la suivante:

- la transition peut être tirée au plus tôt à $\min(x, a)$
- la transition t est franchi au plus tard à b .

Par exemple, si nous avons $(x, y) = (10, 40)$ et $(a, b) = (15, 30)$ alors la transition t est tirable dans l'intervalle $[10, 30]$, en particulier la transition t peut être tirée à l'instant relatif 10 alors que cette date n'est pas dans l'IVT de l'arc maître.

7.2.1.4 La Synchronisation de Type « Et-Maître »

La synchronisation de type « et-maître » induit une synchronisation logique telle qu'un lien n'est activable que lorsque toutes les présentations associées à ce lien ont atteint la borne inférieure de leur intervalle de validité temporelle. L'utilisateur doit donc voir un composant pendant une durée minimum avant de pouvoir activer un lien de type « et-maître » « ancré » dans ce composant. Ainsi dans l'exemple de la Figure 6.10 la transition t , de type « et-maître », sera franchie:

- au plus tôt à l'instant $\max(a, x)$
- au plus tard à l'instant $\max(b, \max(x, a))$

Par exemple, si nous avons $(x, y) = (20, 35)$ et $(a, b) = (15, 30)$ alors la transition est tirable dans l'intervalle temporel $[20, 30]$. Cela signifie en particulier que même si la condition logique associée au lien est satisfaite à l'instant relatif 15 le lien ne sera pas activé avant l'instant relatif 20.

7.2.1.5 La Synchronisation de Type « Maître-Faible »

La synchronisation de type « maître-faible » induit une synchronisation logique telle qu'un lien ne peut être activé que si la condition logique associée au lien est satisfaite et si au moins un de ses composants « sources » est terminé. Ce type de synchronisation introduit donc une forme d'indéterminisme de synchronisation borné inférieurement par la borne inférieure de l'IVT associé au lien. Ainsi dans le cas de l'exemple illustré par la Figure 6.10, la transition t de type « maître-faible » sera tirable:

- au plus tôt à l'instant a
- au plus tard à l'instant $\max(b,y)$

Par exemple si nous avons $(x,y)=(20,35)$ et $(a,b)=(15,30)$, alors la transition t est tirable dans l'intervalle temporel $[15,35]$.

7.3 Liens Synchrones et Liens Asynchrones

Le modèle RdPFT permet d'introduire les notions de lien synchrone et de lien asynchrone. On appelle liens synchrones les liens qui apparaissent de façon synchrone avec leurs composants « sources » (i.e. les dates de sensibilisation de l'arc modélisant le lien et celles des arcs modélisant les composants sources sont identiques). Inversement un lien asynchrone peut être sensibilisé à une date différente de celle de ses composants « sources ».

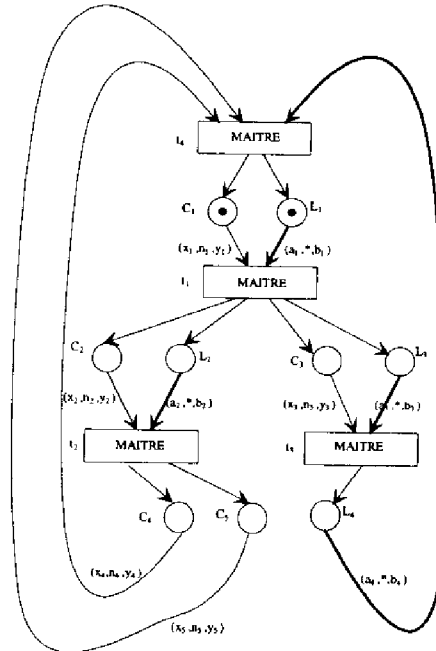


Figure 6.8: Modélisation de liens synchrones et de liens asynchrones.

Par exemple, les liens (L_1, t_1) , (L_2, t_2) et (L_3, t_3) du RdPFT représenté dans le Figure 6.8 sont des liens synchrones (i.e. la date de sensibilisation de (L_2, t_2) est la même que celle de (C_2, t_2)). Par contre le lien (L_4, t_4) est asynchrone car il n'est pas sensibilisé en même temps que les composants (C_4, t_4) et (C_5, t_4) (dont la sensibilisation résulte du

tir de la transition t_2) mais suite au tir de la transition t_3 . Ainsi le parcours d'une partie du document (dans cet exemple les composants C_4 et C_5) est conditionné par le parcours d'une autre partie du document (dans cet exemple le composant C_3).

7.4 Modélisation de l'Indéterminisme

L'indéterminisme est une caractéristique fondamentale des systèmes hypermédias. Il s'exprime sous la forme de choix multiples d'activations de liens à partir d'un noeud du réseau hypermédia. L'indéterminisme inhérent aux systèmes hypermédias peut être aisément modélisé par l'intermédiaire du modèle RdPFT. A titre d'exemple, la Figure 6.9 illustre un RdPFT qui modélise deux liens, (i.e. les arcs (L, t_1) et (L, t_2)) ayant le même composant « source » (i.e. (C, t)). La place « L » de type lien modélise un menu permettant d'activer plusieurs liens. On remarquera que chaque lien a son propre intervalle de validité temporelle.

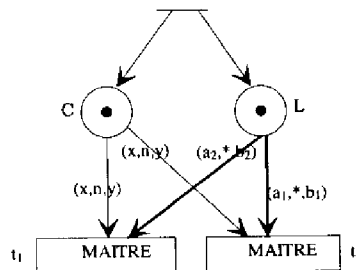


Figure 6.9: Modélisation de l'indéterminisme.

8. Vision Globale de la Modélisation des Systèmes Hypermédias

La modélisation des composants hypermédias à l'aide du modèle RdPFT induit le modèle hiérarchique à trois niveaux illustré par la Figure 6.11.

Le Niveau de Synchronisation Logique

Le plus haut niveau de la Figure 6.11, appelé niveau de synchronisation logique, est dédié à la spécification des contraintes de synchronisation logique dans les systèmes hypermédias. Ce niveau spécifie, en terme de liens hypermédias, la sémantique de parcours du document hypermédia. Dans ce niveau, les places et les arcs associés représentent des composants atomiques (pour les places de type atomique), des composants composites (pour les places de type composite) ou des parties de composants (atomiques ou composites) « adressées » par une ancre, ou bien un lien (pour les places de type lien). Ainsi ce niveau de spécification n'adresse pas seulement la modélisation des liens entre les composants mais aussi la modélisation des possibilités de navigation à l'intérieur d'un composant atomique ou composite. De plus ce niveau explicite les contraintes de synchronisation « à granularité maximale » intrinsèques aux composants. Il est important de remarquer que les RdPFT générés à ce niveau ne doivent satisfaire aucune contrainte structurelle. Ainsi, ces RdPFT ne sont pas nécessairement structurés, et en particulier ils peuvent mettre en oeuvre de l'indéterminisme et des cycles.

Le niveau de Synchronisation Composite

Le niveau suivant (i.e. le niveau synchronisation composite) s'attache à la spécification des contraintes de synchronisation temporelle à l'intérieur des composants composites. Les RdPFT de ce niveau sont constitués de places de type atomique ou, récursivement, de type composite. Ainsi une place d'un RdPFT (i.e. un sous-réseau) appartenant au niveau de synchronisation composite peut être associée à un autre RdPFT de ce même niveau. Cette définition récursive permet de modéliser la structure hiérarchique des composants composites telle qu'elle est définie par le modèle Dexter. Dans le niveau de synchronisation composite, les composants (i.e. les scénarios multimédias) sont modélisés sous la forme de RdPSFT associés à une place composite appartenant au RdPFT du niveau de synchronisation logique ou à un RdPSFT du niveau de synchronisation composite. Les RdPSFT sont utilisés à ce niveau car, comme souligné par le Chapitre 5, ils possèdent de bons pouvoirs de modélisation, d'expression et d'analyse pour la spécification de scénarios de synchronisation multimédia. En particulier les RdPSFT ont de « bonnes » propriétés structurelles qui permettent d'appliquer des méthodes de vérification pour s'assurer de la cohérence temporelle des scénarios de synchronisation multimédia constitutifs des composants composites.

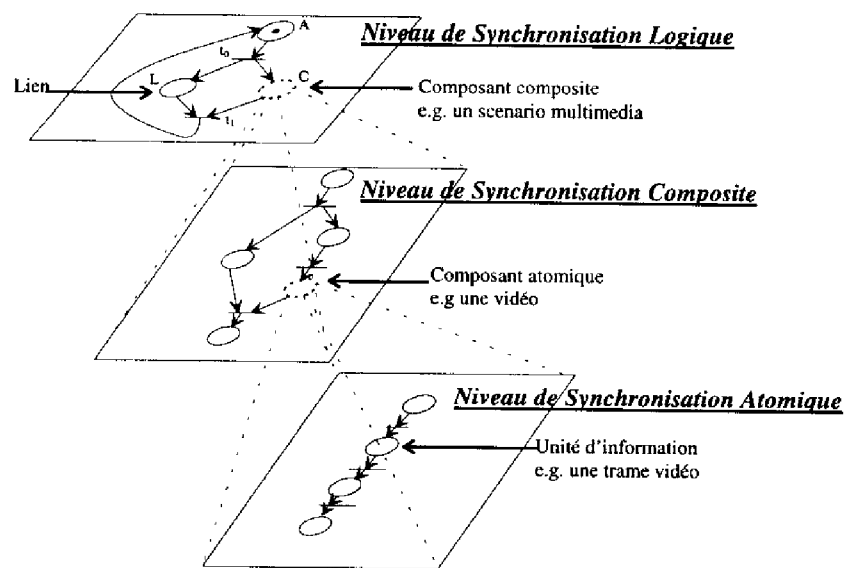


Figure 6.11: Niveaux de synchronisation dans un système hypermédia

Le niveau de Synchronisation Atomique

Le troisième niveau (i.e. le niveau de synchronisation atomique) s'attache à la spécification des contraintes de synchronisation temporelle à l'intérieur des composants atomiques. En particulier, ce niveau est dédié à la spécification des contraintes de synchronisation intra-flux et à la spécification des schémas de synchronisation continue qui, dans le cadre du modèle Dexter, sont du domaine du « within-layer ». Ce niveau, tout comme le niveau de synchronisation composite, est constitué de RdPSFT. Cependant les RdPSFT du niveau de synchronisation atomique sont constitués de schémas de synchronisation plus élémentaires et à plus fine granularité que les RdPSFT du niveau de synchronisation composite.

9. Les Réseaux de Petri Hiérarchisés à Flux Temporels

9.1 Définition

Le modèle de synchronisation hiérarchique à 3 niveaux précédemment introduit induit une hiérarchie de RdPFT que nous appellerons Réseau de Petri Hiérarchisé à Flux Temporel (RdPHFT) [SS-SW95].

Définition 6.1: Un RdPHFT est un tuple $H=(R,S,Pen,FS,Fen)$ tel que:

- $R=(P_r, T_r, Pré_r, Post_r, M_r, Syn_r, AM_r, Tp_r)$ est un RdPFT étendu à l'aide d'une fonction de typage des places Tp_r . Autrement dit:
 - $(P_r, T_r, Pré_r, Post_r, M_r, Syn_r, AM_r)$ définit le RdPFT origine de la hiérarchie (i.e. le RdPFT spécifiant le niveau de synchronisation logique). Ce RdPFT est appelé réseau racine du RdPHFT et sera supposé « sain » (i.e. $\forall p \in P_r, M_r(p) \leq 1$).
 - $Tp_r: P_r \rightarrow \{\text{atomique, composite, lien}\}$ est la fonction de typage des places du réseau racine. Les places de type « lien » sont représentées graphiquement en trait gras.
- $S=\{S_i / i \in I\}$ est un ensemble fini de RdPSFT constitutifs dans le cas de la modélisation de systèmes hypermédiés des niveaux de synchronisation composite et atomique. Ainsi nous avons:
 - $S_i=(P_i, T_i, Pré_i, Post_i, M_i, Syn_i, AM_i, Tp_i)$ définit un RdPSFT étendu tel que:
 - $(P_i, T_i, Pré_i, Post_i, M_i, Syn_i, AM_i)$ est un RdPSFT,
 - $Tp_i: P_i \rightarrow \{\text{composite, atomique}\}$ est la fonction de typage des places du réseau S_i
 - avec $\forall (i,j) \in I^2 / i \neq j, ((P_i \cup T_i) \cap (P_j \cup T_j)) = \emptyset$
- $Pen \subset P$ avec $P = \bigcup_{i \in I} P_i$, est l'ensemble des places d'entrée des sous-réseaux appartenant à S .
- $FS: C \rightarrow I \cup \{\text{nul}\}$ est la fonction qui associe toute place composite ou atomique avec un élément de S (i.e. un sous-réseau) et où $C = \{p \in P / Tp(p) = \text{composite}\} \cup \{p \in P / Tp(p) = \text{atomique}\}$. De plus cette fonction permet d'associer un sous-réseau « nul » c'est à dire indéfini à une place composite ou atomique. Cette possibilité permet de réaliser une spécification de façon descendante par raffinages successifs ou bien de ne pas détailler les contraintes temporelles de certains composants atomiques ou composites. Tout comme dans le modèle Dexter la seule contrainte sur la hiérarchie est d'induire un graphe orienté acyclique. C'est à dire qu'aucun réseau de la hiérarchie ne doit contenir une place atomique ou composite directement ou indirectement associée à ce réseau. Les places p de H telles que $FS(p) \neq \text{nul}$ sont représentées graphiquement en trait pointillé.
- $Fen: I \rightarrow Pen$ est la fonction qui associe tout élément de S à une place d'entrée.

Cette définition s'efforce d'avoir un caractère général en s'abstrayant le plus possible d'une interprétation « hypermédia » qui n'apparaît que dans les fonctions de typage⁴ associées aux réseaux constitutifs d'un RdPHFT. Ainsi cette définition ne rend pas explicite les trois niveaux de synchronisation résultant d'une « interprétation » hypermédia du modèle RdPHFT. Dans un contexte hypermédia on peut expliciter ces trois niveaux, tout en étant

⁴ Dans un cadre général seule la notion de place composite serait nécessaire (i.e. les notions de places de type atomique ou de type lien seraient superflues).

conforme à la définition générale, en exprimant le couple (R,S) d'un RdPHFT sous la forme d'un triplet $H=(R,C,A)$ tel que R est le RdPFT racine (étendu à l'aide d'une fonction de type de places), C est une liste ordonnée (selon l'ordre lexicographique des identificateurs de places composites) de RdPSFT (étendus) constituant le niveau de synchronisation composite, et A est une liste ordonnée (selon l'ordre lexicographique des identificateurs de places atomiques) de RdPSFT (étendus) attachés au niveau de synchronisation atomique. Par la suite, nous utiliserons cette représentation d'un RdPHFT tout en gardant à l'esprit qu'elle est spécifique d'un contexte de modélisation de systèmes hypermédiés.

9.2 Règles de Tir

Les règles de tir traditionnelles des RdPFT sont étendues pour les RdPHFT en considérant que les réseaux constituant un RdPHFT, noté H, évoluent comme un seul réseau. En particulier, cela signifie que les RdPFT de H partagent la même horloge de simulation. Les règles de tir d'un RdPHFT sont alors définies de la façon suivante:

- les règles de tir fixées pour les RdPFT sont conservées et l'ensemble des RdPFT constitutifs de H évolue comme un RdPFT unique.
- de plus, le tir d'une transition, t, appartenant à l'un quelconque des RdPFT élément de H, ayant des places de type composite ou atomique dans son préfixe ou dans son suffixe s'accompagne, pour ces places, des opérations suivantes:
 - retirer tous les jetons marquant les places associées aux sous-réseaux associés directement ou indirectement aux places atomiques ou composites appartenant au préfixe de t. Autrement dit (en supposant que le réseau racine est sain):

$$\forall p \in P_x / (\text{Pré}(p,t) \neq 0) \wedge ((\text{Tp}(p)=\text{composite}) \vee (\text{Tp}(p)=\text{atomique})) \wedge (\text{FS}(p) \neq \text{nul})),$$

retirer(FS(p));

$$\forall p \in P_x / (\text{Pré}(p,t) \neq 0), M_x(p)=0$$

avec $X=I \cup \{r\}$ et $x \in X$ et la fonction récursive « retirer » définie de la façon suivante:

Fonction retirer(j)

$$\forall p \in P_j / ((\text{Tp}(p)=\text{composite}) \vee (\text{Tp}(p)=\text{atomique})) \wedge (\text{FS}(p) \neq \text{nul})), \text{retirer}(\text{FS}(p));$$

$$\forall p \in P_j, M_j(p)=0;$$

fin;

- mettre un jeton dans toutes les places d'entrée des sous-réseaux associés aux places composites ou atomiques appartenant au suffixe de t. Cette opération peut être exprimée formellement de la façon suivante:

$$\forall p \in P_x / (\text{Post}(p,t) \neq 0) \wedge ((\text{Tp}(p)=\text{composite}) \vee (\text{Tp}(p)=\text{atomique})) \wedge (\text{FS}(p) \neq \text{nul})),$$

marquer(FS(p));

$$\forall p \in P_x / (\text{Post}(p,t) \neq 0), M_x(p)=1;$$

où « marquer » est la fonction récursive définie comme il suit:

Fonction marquer(j)

```

Si ( (Tp(Fen(p))=composite)∨ (Tp(Fen((p))=atomique)) ∧(FS(Fen(p)) ≠nul) alors
marquer(FS(Fen(p)));
Mi(Fen(p))=1;
fin;

```

Ces règles de tir étendues permettent de modéliser facilement les interruptions de scénarios multimédias. Les pouvoirs de modélisation et d'expression du modèle RdPHFT pour la spécification des événements asynchrones apporte une contribution importante par rapport à la technique reposant sur les arcs d'activation proposée par [Zub80] et utilisée dans [PR93] pour la modélisation des interactions utilisateurs dans les applications multimédias. En effet, par définition, les arcs d'activation doivent être associés à toutes les transitions qui ont dans leur préfixe des événements susceptibles d'être interrompus. Une telle approche pour la modélisation des interruptions induit un accroissement important de la complexité structurelle des réseaux. De plus cette complexité structurelle est susceptible d'induire une diminution du pouvoir d'analyse du modèle ainsi étendu. Ainsi, en effectuant le retrait des jetons dans les sous-réseaux associés à une place atomique ou composite, les règles de tir du modèle RdPHFT permettent de modéliser facilement les événements asynchrones qui sont générés suite à une activation de lien et qui sont à la base du comportement dynamique et de la sémantique d'un document hypermédia. Autrement dit les capacités d'abstraction offertes par la notion de place composite ou atomique permettent une représentation synthétique des événements asynchrones.

9.3 Etat d'un RdPHFT

La notion d'état d'un RdPFT a été introduite dans le chapitre 4 afin de permettre la simulation formelle du comportement dynamique d'un système hypermédia. De même la notion d'état d'un RdPHFT serait utile pour simuler formellement le comportement dynamique d'un système hypermédia. Or nous avons vu qu'un RdPHFT peut être considéré comme un ensemble de RdPFT; il apparaît donc que la notion d'état d'un RdPHFT résulte d'une extension immédiate de la notion d'état d'un RdPFT. En effet l'état d'un RdPHFT peut être défini comme l'ensemble des états des RdPFT le constituant.

Définition 6.2: L'état d'un RdPHFT $H=(R,S,Pen,FS,Fen)$ est donné par un couple $S=(M,I)$ tel que:

- M est un couple $M=(M_r,(M_i)_{i \in I})$, et
- I est un couple $I=(I_r,(I_i)_{i \in I})$

Avec $\forall j \in I, S_j=(M_j,I_j)$ l'état courant du RdPSFT S_j , et $S_r=(M_r,I_r)$ l'état courant du RdPFT racine R .

En considérant que tous les RdPFT constituant un RdPHFT évoluent de façon synchrone comme un seul réseau, les règles d'évolution entre états d'un RdPHFT résultent alors d'une extension immédiate des règles d'évolution entre états d'un RdPFT.

On peut noter que cette définition d'état tout comme la définition d'un RdPHFT s'attache à être la plus générale possible. Si l'on veut expliciter les trois niveaux de synchronisation intrinsèques aux systèmes hypermédiés, cette notion d'état peut être exprimée en considérant les éléments M et I de l'état S d'un RdPHFT comme des triplets et non comme des couples.

9.4 Modélisation et Simulation des Événements Asynchrones

Considérons le RdPHFT illustré par la Figure 6.12. Ce réseau, noté H, met en oeuvre une simple hiérarchie à deux niveaux, constituée d'un RdPFHT racine, noté R (Figure 6.12-a), et de deux RdPSFT, notés N_1 (Figure 6.12-b) et N_2 (Figure 6.12-c) appartenant au niveau de synchronisation composite. Les deux sous-réseaux N_1 et N_2 sont respectivement associés aux places composites C_1 et C_2 de R. Ainsi ce RdPHFT peut être décrit synthétiquement par le triplet $H=(R,(N_1,N_2))$, dans lequel le troisième élément est vide (i.e. il n'y a pas de spécification appartenant au niveau de synchronisation atomique, à ce titre ce triplet sera considéré comme un couple dans la suite).

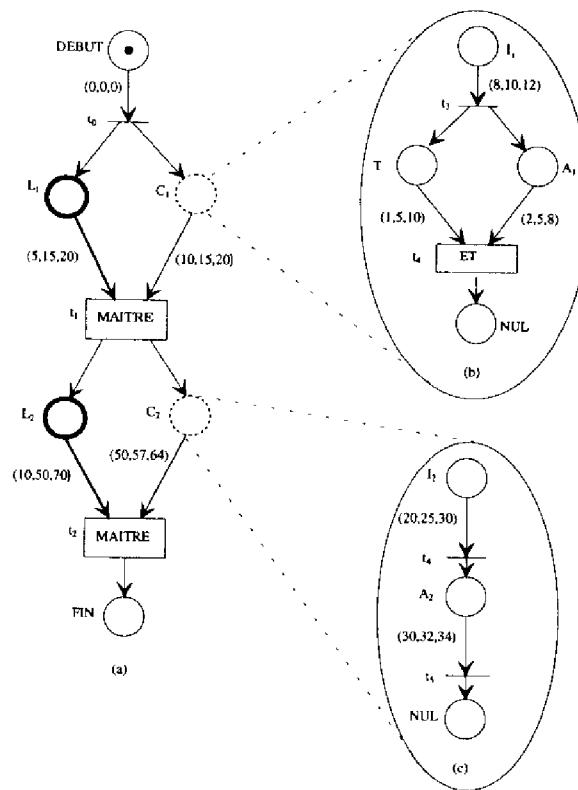


Figure 6.12: Un Réseau de Petri Hiérarchisé à Flux Temporels

L'état initial $S_0=(M_0,I_0)$ de H est donné par:

- Le marquage initial $M_0=(M_{0r},(M_{01},M_{02}))$ qui est la réunion des états de R, N_1 and N_2 . Ainsi, en considérant que les places n'apparaissant pas dans un marquage ont un marquage nul, nous avons:
 - $M_{0r}=(DEBUT(1))$, avec DEBUT(1) signifiant que la place DEBUT possède 1 jeton.
 - $M_{01}=()$
 - $M_{02}=()$

- Le triplet $I_0=(I_{0r},(I_{01},I_{02}))$, où I_{0r} , I_{01} and I_{02} sont respectivement les listes des Intervalles Dynamiques de Validité Temporelle (IDVT) des arcs sensibilisés dans R, N_1 et N_2 . Nous avons donc:

- $I_{0r}=(0,0,0)$
- $I_{01}=()$
- $I_{02}=()$

Suite au tir de la transition t_0 , à partir de l'état S_0 à l'instant relatif $\theta=0$, H atteint l'état $S_1=(M_1,I_1)$, tel que:

- $M_1=((C_1(1),L_1(1)),(I_1(1),()))$
- $I_1(((10,15,20),(5,15,20)),((8,10,12),()))$

A partir de l'état S_1 , la transition de type « maître » t_1 de R doit être tirée dans l'IDVT de l'arc « maître » (L_1,t_1) (i.e. $(5,15,20)$). Ainsi, en accord avec les règles de tir des RdPFT, t_1 doit être tirée à partir de l'état S_1 à l'instant relatif θ tel que $5 \leq \theta \leq \min_i(M_i) = 12$ où les M_i sont les bornes supérieures des intervalles de tir des transitions sensibilisées dans l'état S_1 . En particulier la transition t_1 est tirable à l'instant relatif $\theta=5$ alors que la place I_1 du sous réseau N_1 est encore marquée. Cela signifie que le tir de t_1 , à l'instant relatif $\theta=5$, entraîne le retrait du jeton marquant la place I_1 dans le sous-réseau N_1 . Ce retrait entraîne ainsi le « saut » des places T_1 et A_1 appartenant à ce sous-réseau. Ainsi, suite au tir de la transition t_1 à l'instant relatif $\theta=5$ depuis l'état S_1 le réseau H atteint l'état $S_2=(M_2,I_2)$ tel que:

- $M_2=((C_2(1),L_2(1)),((),I_2(1)))$
- $I_2(((50,57,64),(10,50,70)),((),(20,25,30)))$

On remarquera que le marquage du sous-réseau N_1 est maintenant nul.

Les trois réseaux évoluent de façon synchrone. Ainsi, la transition t_4 de N_2 est tirable à partir de l'état S_2 à l'instant relatif θ tel que: $20 \leq \theta \leq \min_i(M_i) = 30$. Si la transition t_4 est tirée à partir de l'état S_2 à l'instant relatif $\theta=30$, alors H atteint le nouvel état $S_3=(M_3,I_3)$ tel que:

- $M_3=((C_2(1),L_2(1)),((),(A_2(1))))$
- $I_3(((20,27,34),(0,20,40)),((),((30,32,34))))$.

On remarquera que les règles d'évolution entre états des RdPFT ont été étendues aux RdPHFT. Ainsi, les bornes des IDVT⁵ de S_2 ont été réduites d'une valeur $\theta=30$.

La transition t_2 de type « maître » (ayant (L_2,t_2) comme arc « maître ») est tirable depuis S_3 à l'instant relatif θ vérifiant, $0 \leq \theta \leq \min_i(M_i) = 34$. Si t_2 est tirée à l'instant relatif $\theta=20$ alors H atteint l'état $S_4=(M_4,I_4)$ tel que:

⁵ Dans le cas d'arcs sensibilisés dans S_2 et qui ne sont pas associés à une place appartenant au préfixe de t_4

- $M_4 = ((\text{FIN}(1)), ((), ()))$
- $I_4 = ((), ((), ()))$

Cet état est l'état final car il n'y a plus de transition tirable.

Ainsi le modèle RdPHFT de par sa capacité à modéliser les événements asynchrones susceptibles d'interrompre des traitements non élémentaires lève la principale limitation des RdPFT, à savoir la nécessité d'avoir recours à une sémantique de synchronisation de type « best-effort » ne garantissant pas dans tous les cas le respect des contraintes temporelles de l'unité de synchronisation que l'on souhaite privilégier en cas de désynchronisation.

10. Exemples

10.1 Une Solution à la « Lip-synchronization » à l'Aide des RdPHFT

Nous présentons ici, à l'aide du modèle RdPHFT, une solution complète au problème de « lip-synchronization » qui a été partiellement résolu dans le chapitre 4 (section 6) avec le modèle RdPFT.

Cette deuxième solution utilise les possibilités de hiérarchisation fournies par le modèle RdPHFT. Contrairement à la solution proposée dans le chapitre 4 celle-ci respecte rigoureusement l'ensemble des contraintes de synchronisation souhaitées. Cette spécification est réalisée à partir d'un RdPHFT comprenant 5 RdPFT et 3 niveaux hiérarchiques (Figure 6.13). Le plus haut niveau de la hiérarchie, illustré par la Figure 6.13-a, spécifie les contraintes de synchronisation inter-flux entre le flux audio modélisé sous la forme de la place atomique⁶ A_0 et le flux vidéo modélisé par la place atomique V_0 . Ce réseau spécifie en particulier le contrôle qui doit être mis en oeuvre pour assurer que le flux vidéo n'ait pas un retard supérieur à 150ms par rapport au flux audio.

Ce contrôle doit être appliqué avec une granularité maximale de 30 unités de synchronisation vidéo (ou 40 unités de synchronisation audio). En effet la gigue maximale des unités de synchronisation vidéo étant de 5ms, et les unités de synchronisation audio ne subissant aucune gigue, une dérive maximum de 150 ms peut être obtenue suite au traitement de 30 unités de synchronisation vidéo. Ainsi le RdPFT de la Figure 6.14-a spécifie qu'un point de synchronisation de type « maître », avec l'unité de synchronisation audio comme arc « maître », est placé toutes les 40 unités de synchronisation audio (i.e. toutes les 1200 unités de temps) de façon à recaler le flux vidéo sur le flux audio. Cette resynchronisation entre le flux audio et le flux vidéo est susceptible d'induire une accélération du flux vidéo réalisée en « sautant », jusqu'à 5 unités de synchronisation vidéo dans le cas d'un retard maximum.

La place abstraite V_0 est spécifiée par l'intermédiaire du sous-réseau illustré par la Figure 6.13-b. Ce RdPFT explicite le contrôle de l'avance du flux vidéo par rapport au flux audio. Etant donnée la gigue des unités de synchronisation vidéo (i.e. 5ms), le contrôle de la dérive du flux vidéo doit être réalisé toutes les 3 unités de synchronisation vidéo. La fonction de contrôle est modélisée sous la forme de places notées C_0 et C_1 . La

⁶ La distinction entre place atomique et place composite n'est pas intrinsèque au modèle RdPHFT, elle résulte d'une « interprétation hypermédia » ce modèle. Ainsi les places V_0 et A_0 sont qualifiées d'atomiques car le modèle Dexter considère ces composants comme tels.

resynchronisation est réalisée à partir de transitions de type «et-faible» induisant éventuellement un ralentissement du flux vidéo par rapport au flux audio (dont les caractéristiques sont modélisées par les places de contrôle C_0 et C_1) afin de le recaler sur le flux audio.

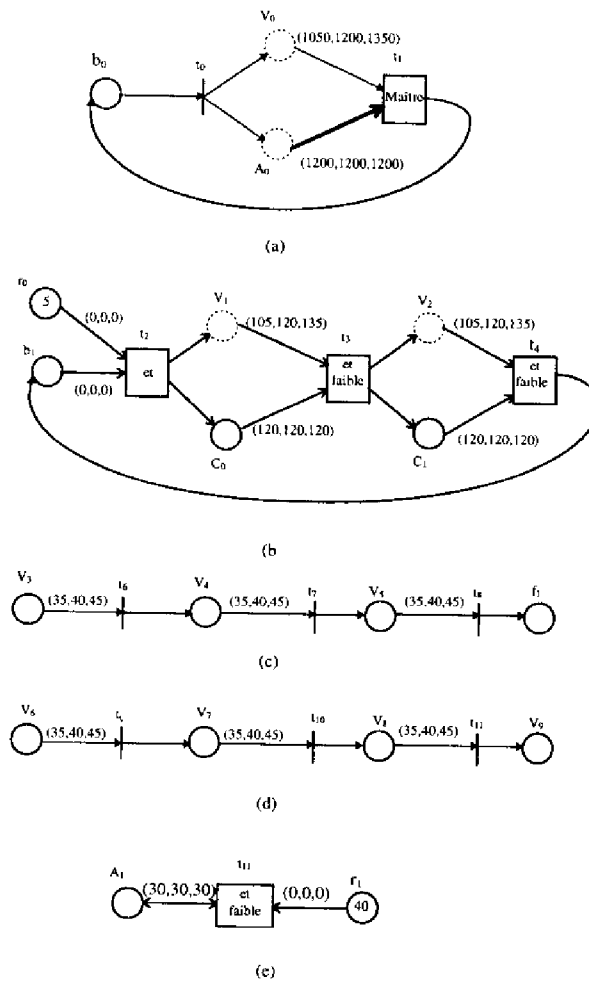


Figure 6.13: Modélisation d'un schéma de « lip-synchronization » à l'aide du modèle RdPHFT. (a) La racine de la hiérarchie de réseaux spécifiant en particulier le contrôle du retard de la vidéo par rapport à l'audio. (b) Spécification de la place abstraite V_0 . Ce niveau décrit le contrôle de l'avance du flux vidéo par rapport à l'audio. (c) et (d) Spécification des places abstraites V_1 et V_2 décrivant les contraintes de synchronisation intra-flux vidéo. (e) Spécification de la place abstraite A_0 décrivant les contraintes de synchronisation intra-flux audio.

Les places atomiques V_1 et V_2 sont spécifiées par les RdPFT illustrés respectivement par les Figures 6.13-c et 6.14-d. Ces RdPFT, décrivent les contraintes de synchronisation intra-flux vidéo. La place atomique A_0 est associée quant à elle au sous-réseau illustré par la Figure 6.13-e. Ce RdPFT spécifie les contraintes de synchronisation intra-flux audio.

10.2 Une Application Hypermédia

Dans cet exemple nous allons mettre en oeuvre le modèle RdPHFT pour la modélisation d'un environnement hypermédia permettant une visite guidée des groupes de recherche d'une Grande Ecole. Cet environnement hypermédia est modélisé par le RdPHFT illustré par la Figure 6.14.

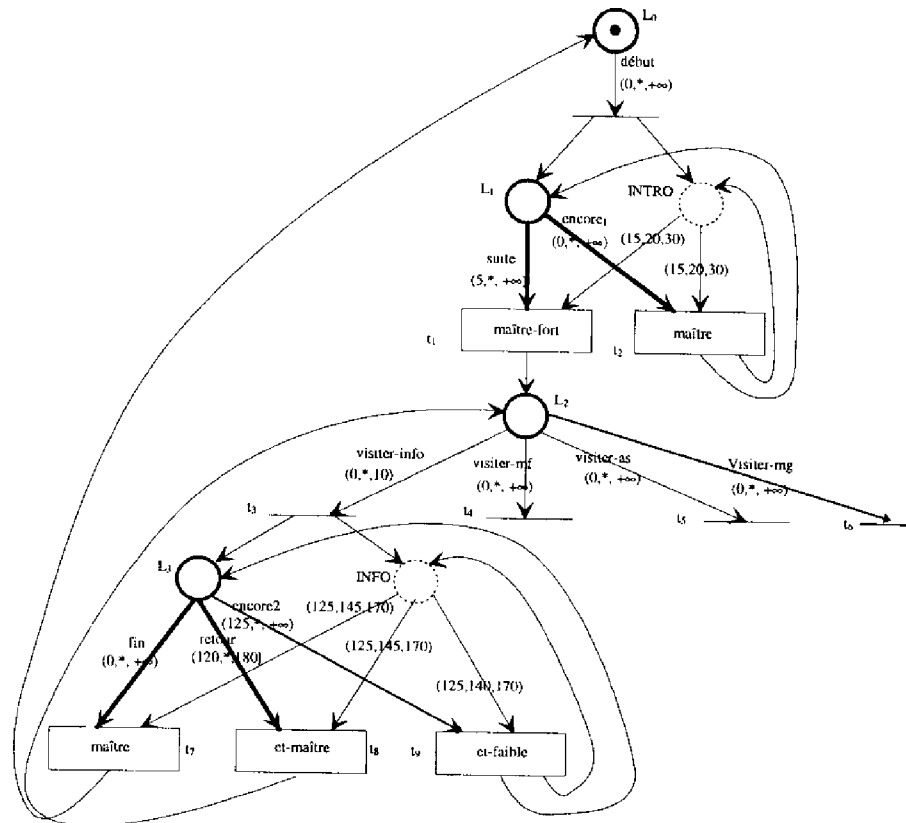


Figure 6.15: Un RdPHFT modélisant une application hypermédia permettant de visiter les laboratoires de recherche d'une Grande Ecole

Cette application hypermédia débute par la présentation d'un bouton (modélisé par la place L_0) permettant d'activer un lien (modélisé par l'arc « début »). L'intervalle d'activation de ce lien est $\{0, \infty\}$; ainsi seule une interaction utilisateur (ou plus généralement la réalisation de la condition associée à cet arc) permettra d'activer le lien « début » de façon à déclencher la présentation d'un scénario multimédia présentant l'Ecole. Ce scénario introductif est modélisé par la place composite INTRO qui est telle que tous les arcs a_i issus de cette place vérifient $IVT(a_i) = (15, 20, 30)$ (i.e. la place INTRO modélise un scénario multimédia ayant pour durées minimale, nominale et maximale de présentation respectivement 15, 20 et 30 unités de temps). Parallèlement à la présentation de ce scénario d'introduction, l'utilisateur a la possibilité d'activer deux liens, modélisés par les arcs « suite₁ » et « encore₁ », lui permettant respectivement de passer à la suite de la présentation ou de relancer l'introduction. Le lien « suite » est l'arc « maître-fort » de la transition t_1 ; à ce titre l'activation de ce lien, à

l'intérieur de son intervalle de validité temporelle (i.e. $(5, *, \infty)$), permet d'interrompre l'introduction et de passer à la suite de la présentation. Le lien « suite₁ » n'est activable qu'à partir de 5 unités de temps après le début de l'introduction; ainsi l'utilisateur est obligé d'assister à la première partie de l'introduction (dont la connaissance a été jugée indispensable) avant de pouvoir passer à la suite. Le lien « encore₁ » est l'arc « maître » de la transition t_2 , l'intervalle de validité temporelle associé à cet arc (i.e. $(0, *, \infty)$) permet à l'utilisateur d'interrompre à tout instant la présentation de l'introduction de façon à la rejouer depuis le début. Remarquons que si l'utilisateur n'active aucun de ces deux liens, la transition t_1 , de type « maître-fort », sera automatiquement tirée lorsque la présentation de l'introduction sera terminée (i.e. dans l'intervalle $[15, 30]$).

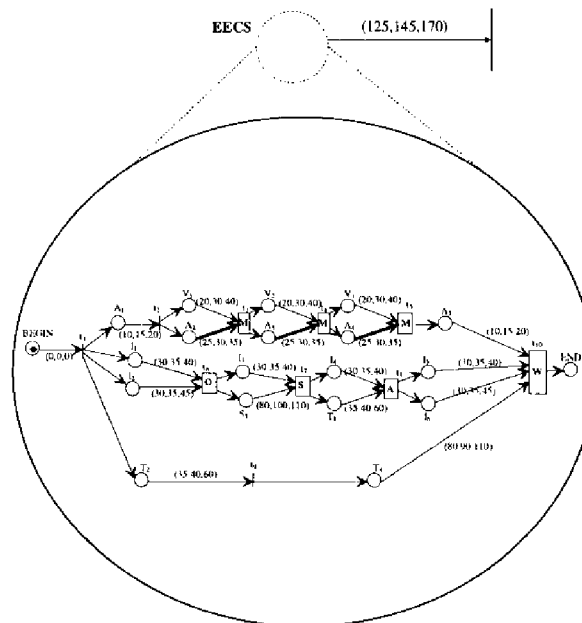


Figure 6.15: La place composite INTRO et son RdPSFT sous-jacent.

Suite au tir de la transition t_1 , l'utilisateur peut activer un des quatre liens, respectivement modélisés par les arcs « visiter-info », « visiter-mf », « visiter-as » et « visiter-mg », qui lui sont proposés par l'intermédiaire du menu (i.e. bouton multi-choix) modélisé par la place L_2 . Le lien « visiter-info » ayant $(0, *, 10)$ pour intervalle de validité temporelle sera automatiquement activé au bout de 10 unités de temps si l'utilisateur n'a pas activé explicitement un des trois liens. En ce sens le lien « visiter-info » est privilégié par rapport aux deux autres liens « visiter-mf » et « visiter-as ». Suite à l'activation implicite ou explicite du lien « visiter-info », un scénario multimédia (modélisé par la place composite INFO) décrivant les activités du département de recherche en Informatique est présenté à l'utilisateur. Chaque arc issu de la place INFO a le même intervalle de validité temporelle, à savoir l'intervalle de validité temporelle associé au RdPSFT sous-jacent à la place EECS et représenté par la Figure 6.15. On remarquera que ce RdPSFT est celui du scénario de synchronisation multimédia introduit dans le chapitre 4 afin de modéliser la présentation multimédia du département de recherche en informatique. L'intervalle de validité temporelle associé aux arcs issus de la place composite INTRO a été

obtenu en réduisant, comme détaillé dans le chapitre 5 (section 9.2), ce RdPSFT en un arc temporellement équivalent.

L'utilisateur peut agir sur la présentation du département informatique à partir d'un menu modélisé par la place L_3 et permettant d'activer trois liens. Le lien « fin », de type « maître » et d'intervalle de validité temporelle $(0, *_{\infty})$, permet à tout instant d'interrompre la présentation et de revenir au début du document hypermédia. Le lien « retour » de type « et-maître » et d'intervalle de validité temporelle $(120, *, 180)$ permet d'interrompre la présentation multimédia de façon à revenir une étape en arrière, c'est à dire au niveau de la présentation du menu L_2 . Ce lien est présenté à la date relative 120 mais n'est activable (par définition de la synchronisation de type « et-maître ») que lorsque la présentation multimédia associée atteint la borne inférieure de son intervalle de validité temporelle (i.e. : à l'instant relatif $125 = \max(120, 125)$). Ce lien sera explicitement activé à l'instant relatif $180 = \max(125, 180)$ si aucun autre lien n'a été activé. Le lien « encore₂ », de type « et-faible » et d'intervalle de validité temporelle $(125, *, \infty)$, permet, suite à son activation et à la terminaison de la présentation multimédia, de présenter une nouvelle fois les activités du département informatique. En accord avec la sémantique de synchronisation « et-faible », ce lien peut être activé dans l'intervalle temporel $[125, \max(170, +\infty)]$. L'utilisateur ne peut ainsi relancer la présentation que lorsque celle-ci est terminée. Dans le cas le plus défavorable (i.e. la présentation multimédia dure 170 unités de temps) l'utilisateur dispose d'un délai de 10 unités de temps pour activer le lien « encore₂ » avant que le lien « retour » ne soit automatiquement activé.

11. Conclusion

Nous avons introduit dans ce chapitre un nouveau modèle formel nommé RdPHFT permettant de spécifier précisément, complètement et de façon unifiée (i.e. par l'intermédiaire d'arcs temporisées) les contraintes de synchronisation logique et temporelle qui sont fondatrices de la sémantique des systèmes hypermédiés. Les contributions et les éclaircissements apportés par ce modèle sont multiples. En particulier on peut souligner les apports suivants:

- Le modèle RdPHFT a un pouvoir de modélisation permettant une spécification aisée et unifiée des composants (i.e. les liens les composants atomiques et composites) d'un système hypermédia.
- Le modèle RdPHFT apporte des éclaircissements sur la notion de synchronisation hypermédia. En effet, le modèle RdPHFT donne, à l'aide de règles de synchronisation permettant de combiner de façon complète les contraintes de synchronisation temporelle et logique, une sémantique formelle de la notion de synchronisation hypermédia.
- Les règles de tir des RdPHFT associées à leurs possibilités de hiérarchisation permettent une modélisation aisée des événements asynchrones susceptibles d'interrompre à tout instant un traitement multimédia.
- Le modèle RdPHFT a le pouvoir d'expression d'une machine de Turing.
- Les techniques d'analyse développées pour le modèle RdPHFT permettent de vérifier les propriétés logiques et temporelles du système ainsi modélisé.

Chapitre 7

Conclusions et Perspectives

Partant de la considération que les contraintes de synchronisation sont au coeur de la problématique de la conception des systèmes multimédias et qu'à ce jour aucun modèle ne permettait de spécifier à la fois de façon formelle, complète, précise et aisée des scénarios de synchronisation multimédia, nous avons dans un premier temps introduit dans cette thèse un nouveau modèle (i.e. le modèle RdPFT) comblant les limitations des modèles proposés jusqu'alors. Le modèle RdPFT a été conçu selon une approche minimaliste en ayant pour objectif d'offrir le meilleur compromis entre un pouvoir d'expression permettant de spécifier de façon complète et précise, un pouvoir d'analyse offrant des techniques de vérification et de validation performantes et un pouvoir de modélisation permettant une spécification aisée des scénarios de synchronisation.

Le modèle RdPFT a été initialement conçu pour la modélisation de systèmes multimédias. Cependant ce modèle a été développé avec un souci constant de généralité. Ainsi la notion d'unité de synchronisation est apte à décrire les caractéristiques temporelles de tout traitement dans un système faiblement synchrone. De plus, le caractère générique de la notion de système faiblement synchrone permet à partir d'un même formalisme de couvrir tout le spectre partant des systèmes totalement asynchrones (i.e. ayant des durées de traitements non bornées) et allant jusqu'aux systèmes totalement synchrones (i.e. systèmes dans lesquels on fait l'hypothèse que les traitements sont de durée nulle). Ainsi le modèle RdPFT offre un cadre général pour la spécification des contraintes de synchronisation et de temps.

Les contributions apportées par le modèle RdPFT sont nombreuses; en particulier le modèle RdPFT est à notre connaissance le premier modèle proposant une sémantique formelle de la notion de synchronisation dans les systèmes faiblement synchrones. Le pouvoir d'expression du modèle RdPFT, identique à celui d'une machine de Turing, permet de modéliser tout algorithme à partir de ce nouveau formalisme. De plus le modèle RdPFT permet d'exprimer précisément l'indéterminisme temporel et l'indéterminisme de synchronisation intrinsèques aux systèmes faiblement synchrones.

Le modèle RdPFT apporte également une contribution en matière de pouvoir de modélisation. En effet en étant basés sur un modèle temporel reposant sur un temps relatif, les RdPFT comblent par leur caractère structurant,

les limitations des modèles temporels basés sur un temps absolu, tout en offrant, à la différence des extensions temporelles aux réseaux de Petri proposées jusqu'alors, un pouvoir de modélisation permettant de spécifier des scénarios de synchronisation tout aussi aisément qu'à partir du paradigme de composition de flux de la « ligne temporelle ».

De par la possibilité de traduire tout RdPFT en un réseau de Petri temporel, le pouvoir d'analyse du modèle RdPFT hérite du pouvoir d'analyse des réseaux de Petri temporels. En particulier les méthodes d'analyse énumératives proposées dans [BD91] et [BAB93] peuvent être utilisées de façon à mettre en évidence les propriétés logiques et temporelles d'un système modélisé à l'aide du formalisme RdPFT.

Ainsi le modèle RdPFT réunit dans un même formalisme des pouvoirs d'expression, de modélisation et d'analyse qui n'étaient regroupés jusqu'alors dans aucun autre formalisme. Cependant les contributions de ce nouveau formalisme ne s'arrêtent pas là. En effet le modèle RdPFT a été étendu afin de permettre, à partir d'un même formalisme, la modélisation des structures temporelles, logiques et sémantiques des systèmes hypermédias. Cette extension du modèle RdPFT donne lieu au modèle RdPHFT qui offre un cadre unificateur pour la modélisation des structures fondamentales des systèmes hypermédias. En effet à l'aide du modèle RdPHFT les structures temporelles, logiques et sémantiques des systèmes hypermédias peuvent être modélisées de façon uniforme sous forme d'arcs temporels autrement appelés unités de synchronisation. Le modèle RdPHFT propose une extension naturelle des RdPFT par l'intermédiaire de mécanismes de hiérarchisation qui font apparaître un RdPHFT comme un ensemble hiérarchisé de RdPFT. Les RdPHFT bénéficient par là même des techniques de vérification et d'analyse développées pour les RdPFT. Ainsi le modèle RdPHFT participe du même principe minimaliste que le modèle RdPFT et apporte à la modélisation des systèmes hypermédias une contribution en matière de pouvoirs d'expression, de modélisation et d'analyse, du même ordre que celle apportée par les RdPFT pour la modélisation des systèmes multimédias. Ce principe minimaliste vise à proposer, à travers un nombre restreint de concepts, un modèle abstrait permettant une spécification formelle et aisée d'un système hypermédia. Le souci constant de généralisation et d'unification qui a présidé à la définition du modèle RdPHFT, prévient ce nouveau modèle de toute spécificité multimédia/hypersmédia et en fait un modèle unificateur pour la spécification de systèmes « temps-réels ». Le caractère unificateur de ce modèle est dû à un double constat:

- les RdPHFT permettent à partir d'un même formalisme de modéliser de façon uniforme tout le continuum des systèmes « à contraintes temporelles »,
- les composants fondamentaux des systèmes « temps-réels » à savoir les traitements temporellement contraints et les événements asynchrones sont modélisés de façon unifiée sous forme d'arcs temporisés.

De plus les travaux présentés dans [ACCCD94] sur les Protocoles d'Ordre Partiel montrent que le modèle RdPFT/RdPHFT fournit les bases formelles permettant de dériver des paramètres de qualité de service « Transport » à partir d'une qualité de service applicative exprimée sous forme d'un scénario RdPFT. Ainsi le modèle RdPFT fournit un lien entre applications et protocoles de transport multimédias.

Les qualités intrinsèques d'un modèle formel ne justifient pas à elles seules l'utilisation d'une nouvelle technique de spécification. En effet un modèle doit de plus s'accompagner de guides méthodologiques à l'attention de l'utilisateur. Autrement dit les modèles RdPFT/RdPHFT doivent induire un cadre méthodologique guidant et

assistant l'utilisateur dans le processus de spécification d'un système. De plus, le caractère formel des modèles RdPFT/RdPHFT et des techniques de simulation, vérification, analyse qu'ils mettent en jeu, conjugué à la complexité des systèmes qui peuvent être spécifiés à l'aide de ces modèles, rend nécessaire la réalisation d'outils susceptibles d'assister l'utilisateur dans son activité de modélisation. Ainsi les deux sections suivantes vont présenter les contributions des modèles RdPFT/RdPHFT dans le domaine méthodologique et dans celui des outils supports de la modélisation.

1. Méthodologie

Le modèle RdPHFT s'attache exclusivement à la description des structures temporelles, logiques et sémantiques des systèmes hypermédias. Ainsi la modélisation d'un système hypermédia à l'aide du modèle RdPHFT impose au spécifieur de faire abstraction des attributs de présentation spatiale du document. On peut noter cependant que les structures spatiales associées à un document hypermédia sont fortement dépendantes des structures temporelles, logiques et sémantiques composant ce document, et à ce titre peuvent être inférées à partir d'une spécification RdPHFT. Des expérimentations ont permis de quantifier la variabilité temporelle admissible des unités de présentation multimédia [SE93]. Ainsi la détermination des intervalles de validité temporelle associés aux unités de synchronisation de niveau applicatif est facilitée par ces résultats expérimentaux.

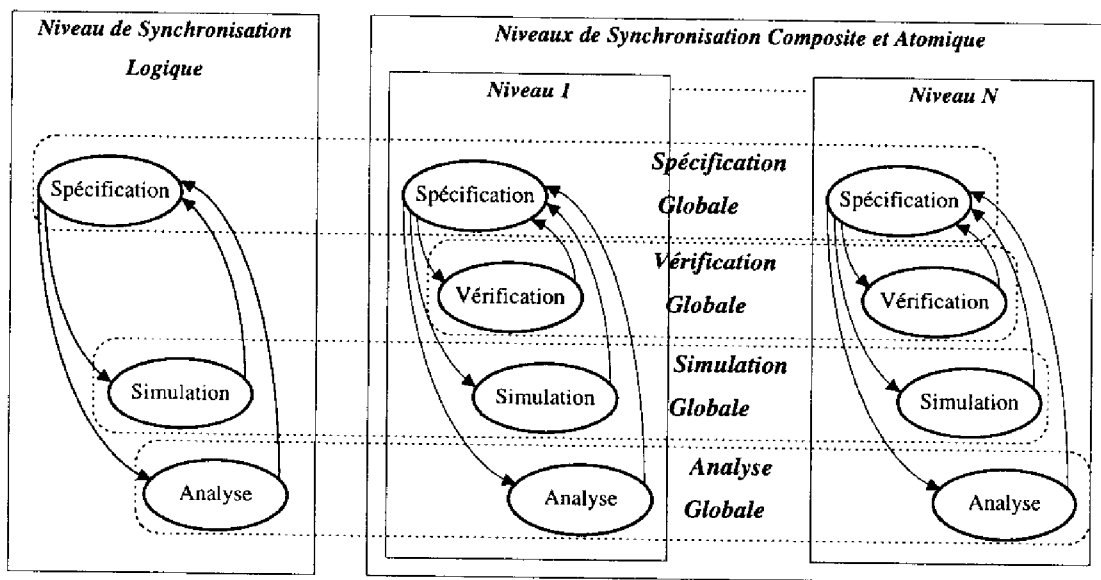


Figure 7.1: Méthodologie de spécification supportée par le modèle RdPHFT

Il n'en est pas de même si l'on souhaite utiliser les RdPHFT pour modéliser les schémas de synchronisation mis en jeu dans les niveaux sous-jacents au niveau applicatif, au sein d'une des étapes constitutives d'un « pipeline » de traitements participant à la mise en oeuvre d'une application multimédia ou hypermédia distribuée. En effet les systèmes d'exploitation, les réseaux et périphériques impliqués dans un tel « pipeline » de traitements ne sont

pas encore en mesure de garantir une certaine Qualité de Service. Dans de tels environnements des mesures expérimentales doivent donc être réalisées afin de déterminer les intervalles de validité temporelle des diverses unités de synchronisation mises en jeu. Les capacités d'abstraction et d'encapsulation du modèle RdPHFT fournissent un support méthodologique permettant de spécifier un système « temps-réel » selon une approche ascendante et/ou descendante [FCSD95]. La première étape de la méthodologie de modélisation induite par le modèle RdPHFT consiste à produire une spécification abstraite de « haut niveau » du système à modéliser (Figure 7.1). Le résultat de cette spécification constitue le RdPFT racine du RdPHFT. Les diverses places composites de ce réseau racine décrivent de façon abstraite le comportement temporel global du système hypermédia. Ce comportement temporel pourra être progressivement détaillé selon une approche descendante basée sur des raffinages successifs de cette spécification initiale. Ainsi dans un premier temps le spécifieur porte son effort sur la modélisation du niveau de synchronisation logique en faisant abstraction des niveaux de synchronisation composite et atomique. D'une façon identique la spécification des schémas de synchronisation du niveau « composite » peut être réalisée en faisant abstraction des contraintes de synchronisation associées au niveau « atomique ». Une telle approche induit un cycle de vie incrémental dans lequel les niveaux supérieurs sont modélisés, simulés, vérifiés et validés avant les niveaux inférieurs.

Inversement une spécification de niveau « N » peut être réalisée avant une spécification de niveau « N-1 » (i.e. de plus haut niveau, voir Figure 7.1). Ainsi les mécanismes modulaires d'encapsulation offerts par le modèle RdPHFT supportent une approche ascendante pour la modélisation d'un système. L'intérêt d'une approche ascendante réside dans le fait qu'elle permet de réutiliser des éléments modulaires de spécification préalablement spécifiés et validés; ces composants de spécification étant associés à des places composites ou atomiques de RdPFT de niveaux supérieurs.

La notion d'intervalle de validité temporelle fournit également un support méthodologique permettant de converger progressivement vers une spécification précise et détaillée du système que l'on cherche à modéliser. En effet, en début du processus de spécification, le spécifieur possède une vision incomplète et imprécise du système qu'il souhaite modéliser. Cette imprécision peut être traduite sous la forme d'intervalles de validité temporelle « grossiers » exprimant des contraintes de synchronisation très lâches, et qui seront raffinées ultérieurement. De même le spécifieur peut décrire initialement des schémas de synchronisation induisant des relations temporelles non déterministes, puis supprimer progressivement ce non-déterminisme, en modifiant les IVT et les types de synchronisation utilisés, au fur et à mesure que sa compréhension du système augmente.

La seconde étape du processus de spécification consiste à vérifier la cohérence temporelle des composants dont le respect des contraintes de synchronisation est le plus critique pour le système hypermédia, à savoir les composants atomiques et composites. Les techniques de vérification, détaillées dans le Chapitre 4, permettent de mettre en évidence les risques de désynchronisation intempestifs susceptibles de causer l'incohérence temporelle de certaines unités de synchronisation. En réponse aux risques de désynchronisation mis en évidence lors de cette étape, l'utilisateur peut:

- changer le type de synchronisation associé aux points de synchronisation inter-flux mis en cause,
- modifier la granularité des schémas de synchronisation inter-flux,

- modifier les intervalles de validité temporelle de certaines unités de synchronisation.

Les délais d'attente conditionnels et la nature des actions correctives ou de substitution devant être effectuées en réponse à des problèmes de désynchronisation peuvent être définis également lors de cette étape [Ste90][SF92].

La troisième étape est consacrée à la simulation du comportement dynamique du système ainsi modélisé. Cette simulation est mise en oeuvre formellement à l'aide de la notion d'état d'un RdPFT. Grâce à la notion d'état le système ainsi modélisé peut être considéré comme un prototype du système hypermédia final. Deux types de simulations peuvent être réalisés lors de cette étape: une simulation automatique ou bien une simulation interactive. La simulation interactive, en faisant participer explicitement le futur utilisateur du système et en lui permettant d'interagir de façon réaliste avec le système, permet d'acquérir des connaissances additionnelles sur le comportement dynamique du système et sur son acceptation par l'utilisateur. La simulation automatique permet quant à elle de réaliser des mesures de performances à partir d'une analyse de traces de simulation.

La quatrième étape de la méthodologie de spécification s'attache à l'analyse des propriétés logiques et temporelles du système ainsi modélisé. Cette étape est réalisée à la suite d'une traduction de la spécification RdPHFT dans le formalisme des Réseaux de Petri Temporels (RdPT). Le RdPT résultant peut être analysé conformément aux méthodes proposées dans [BD91] ou [BAB93]. Dans le cas où le graphe d'état est borné, une analyse exhaustive des propriétés logiques et temporelles du système peut alors être réalisée.

A tout instant pendant le processus de spécification l'utilisateur peut modifier sa modélisation et entamer un nouveau cycle de spécification conformément à un cycle de vie en spirale associant les techniques de spécification, vérification, simulation et analyse qui viennent d'être évoquées. De plus, pour chaque étape, le spécifieur peut adopter un point de vue plus ou moins global, plus ou moins détaillé, en faisant porter cette étape sur tout ou partie de la spécification. Ainsi à l'aide d'une méthode permettant librement une approche descendante, ascendante ou une combinaison des deux, les divers composants d'une spécification s'appuyant sur le modèle RdPHFT peuvent être modélisés de façon indépendante conformément au cycle de vie que nous venons d'évoquer. Globalement le cycle de vie induit par le modèle RdPHFT apparaît donc comme un cycle de vie « à deux dimensions »:

- une dimension mettant en jeu un cycle de vie en spirale contribuant à valider la spécification;
- une dimension mêlant approches ascendante et descendante contribuant à compléter ou raffiner la spécification.

2. Outils

Un atelier de spécification a été développé pour assister le spécifieur dans la mise en oeuvre de la méthodologie introduite dans la section précédente. Une première version de cet atelier, orientée RdPFT et permettant de plus l'utilisation de places abstraites temporellement et structurellement équivalentes à un RdPFT sous-jacent est disponible [SFCD95][FSS-S95]. Cet atelier intègre un éditeur, un simulateur, un vérifieur et un analyseur accessibles à partir d'une interface graphique multi-fenêtrée de type X-windows.

L'éditeur RdPFT assiste le spécifieur dans la construction d'une spécification RdPFT. L'édition d'un RdPFT peut être réalisée selon deux modes:

- un mode structuré qui utilise exclusivement les treize structures de synchronisation déterminées par Allen. L'utilisation de ce mode débouche sur la création de RdPFT structurés.
- un mode libre dans lequel l'utilisateur construit librement un RdPFT indépendamment de toute contrainte structurelle.

Dans chacun de ces deux modes l'utilisateur est guidé à partir de menus déroulants lui permettant de spécifier interactivement et de façon contrôlée les attributs associés aux unités de synchronisation (i.e. les arcs sortant des places) et aux transitions. La construction peut se faire par raffinement de places ou par combinaison de RdPFT à partir de fusions séries ou parallèles de transitions. De plus l'éditeur garantit que toute place abstraite est temporellement et structurellement équivalente au RdPSFT qui lui est associé.

Cet outil offre également à l'utilisateur des possibilités de vérification statique. En effet cet atelier permet de vérifier la cohérence temporelle des RdPSFT en mettant sous forme canonique ces réseaux puis en leur appliquant une analyse temporelle pessimiste. Les unités de synchronisation susceptibles d'être à l'origine de désynchronisations intempestives sont alors mises en évidence graphiquement au niveau du RdPSFT initial. Ainsi à partir des informations retournées par l'outil de vérification l'utilisateur peut, s'il le désire, éditer ce réseau de façon à supprimer tout risque de désynchronisation.

En mettant en oeuvre la notion d'état d'un RdPFT ainsi que les règles d'évolution entre états, l'atelier de spécification de RdPFT permet de simuler le comportement dynamique d'un RdPFT. Afin de déterminer les dates de terminaison des unités de synchronisation, le simulateur permet d'associer des lois de distribution uniformes ou gaussiennes aux intervalles de validité temporelle. Une simulation pas à pas ou avec des points d'arrêt peut alors être réalisée. L'ensemble des événements issus d'une simulation est mémorisé et peut être ultérieurement exploité à des fins d'analyse des performances ou d'étude du comportement dynamique du système ainsi modélisé. Un code de formes et de couleurs associées aux jetons donne à tout instant au cours de la simulation une indication visuelle sur l'état temporel des unités de synchronisation.

3. Réalisations

Afin de démontrer de façon pragmatique les contributions apportées par le modèle RdPFT/RdPHFT, nous avons confronté ce nouveau modèle à la spécification et la conception effective de systèmes multimédias et hypermédias [BCDSS-S94]. On peut citer en particulier les réalisations suivantes:

- Une visioconférence a été développée sur stations de travail Sun équipées de cartes d'acquisition/compression Parallax. Le moteur de synchronisation de cette visioconférence a été conçu à partir d'une spécification RdPFT des contraintes de synchronisation intra et inter-flux devant être suivies par les flux audio et vidéo au niveau des sites émetteurs et récepteurs. Cette spécification implantée à l'aide du multi-threading et des capacités temps réel du système Solaris s'avère donner des résultats tout à fait satisfaisants tant en terme

d'interactivité que de respect des contraintes de synchronisation. [ODS95] explicite de façon détaillée les contributions apportées dans ce domaine par le modèle RdPFT, et souligne la supériorité de l'approche RdPFT par rapport à une approche traditionnelle basée sur des estampilles temporelles sur laquelle résidait jusqu'alors les systèmes de visioconférence développés sur l'Internet.

- Les apports du modèle RdPFT ont été également mis en évidence dans le domaine de la conception de logiciels multimédias d'enseignement assisté par ordinateur. Le modèle RdPFT a été utilisé pour modéliser et concevoir un prototype étendant les fonctionnalités offertes à l'utilisateur par l'environnement VACBI utilisé par Airbus-Training pour la formation des pilotes et du personnel de maintenance des avions Airbus. [OBD95] montre que l'utilisation du modèle RdPFT permet d'étendre l'application VACBI à l'aide de mécanismes de synchronisation assurant un meilleur impact pédagogique tout en tenant compte du caractère faiblement synchrone des systèmes supportant traditionnellement des applications d'enseignement assisté par ordinateur. Plus généralement [OBD95] montre que le modèle RdPFT peut être utilisé afin de modéliser et concevoir des applications de télé-enseignement.
- Les apports du modèle RdPHFT pour la modélisation d'applications hypermédias ont été confrontés à la spécification d'un système hypermédia industriel. Le système VACBI a servi également de base à cette étude. [SWD95] montre que le modèle RdPHFT permet efficacement de modéliser, simuler et analyser pleinement une application hypermédia complexe. De plus [SWD95] montre que le modèle RdPHFT permet, à partir des structures logiques, temporelles et sémantiques qu'il propose, d'améliorer de façon importante la qualité des systèmes hypermédias ainsi que d'étendre à partir de mécanismes innovants et efficaces la notion de système hypermédia.

4. Perspectives

Les actions d'études et développements relatives au modèle RdPFT et à ses applications vont être poursuivies selon trois axes:

- un axe théorique lié au modèle et s'attachant à définir pour celui-ci une méthode d'analyse énumérative permettant de générer le graphe d'états d'un RdPFT (et par là même celui d'un RdPHFT). L'approche prônée dans cette thèse pour réaliser l'analyse énumérative d'un RdPFT consiste à traduire dans un premier temps un RdPFT en un RdPT puis de réaliser, à partir des méthodes proposées par [BD91] ou [BAB93] une analyse du RdPT résultant. La principale limitation d'une telle approche indirecte réside dans le fossé conceptuel qu'elle induit entre la spécification RdPFT et l'analyse du RdPT équivalent. Ce fossé conceptuel rend difficile et opaque l'interprétation du graphe d'états en obligeant à considérer des entités qui ne sont pas du même niveau d'abstraction que celles utilisées lors de la spécification. Il serait donc utile de pouvoir réaliser directement l'analyse énumérative d'un RdPFT. Les travaux qui ont été amorcés dans ce domaine nous permettent d'envisager la faisabilité d'une méthode d'analyse basée sur une extension de la notion de classe d'états telle qu'elle a été proposée par [BD91] pour l'analyse des RdPT.

- un axe orienté vers les outils de modélisation et de conception. Dans ce domaine l'atelier de spécification RdPFPT doit évoluer vers le formalisme RdPHFT. De plus une réflexion sur l'adaptation du formalisme RdPHFT au concepteur traditionnel d'une application hypermédia doit être entamée. En effet bien que le formalisme des RdPHFT, par son caractère graphique, unificateur, minimaliste et relativement proche des concepts traditionnellement utilisés dans la communauté hypermédia, puisse prétendre à être directement utilisé par un concepteur d'application multimédia/hypermédia, il semble souhaitable de masquer le formalisme réseau de Petri sous-jacent au modèle à l'aide d'une interface utilisateur plus conviviale. Une réflexion doit donc être entamée sur la nature de l'interface utilisateur susceptible de masquer le formalisme du modèle RdPHFT. L'assistance à l'utilisateur lors de la vérification et de la simulation d'une spécification RdPHFT sont également deux domaines de recherche prospectifs.
- un axe applicatif s'attachant à la conception d'applications multimédias/hypermédiat à partir du modèle RdPHFT. En particulier il est souhaitable de fournir à partir du modèle RdPHFT un cycle de vie raccourci permettant « d'exécuter » des spécifications RdPHFT. Ce cycle de vie raccourci peut être mis en oeuvre à partir d'une approche directe avec l'aide d'un exécutif de synchronisation susceptible d'interpréter une spécification RdPHFT ou bien à partir d'une approche indirecte, suite à une étape de traduction vers un format normalisé de document hypermédia puis avec l'aide d'un exécutif susceptible d'interpréter tout document répondant au format de traduction. La première solution exige le développement d'un exécutif de synchronisation susceptible d'interpréter une spécification RdPHFT étendue, en particulier, par des attributs de présentation spatiaux. Un tel exécutif doit être capable de générer un profil d'ordonnancement d'unités de synchronisation à partir d'une spécification RdPHFT. Si l'on se place dans un environnement faiblement synchrone l'indéterminisme temporel intrinsèque à de tels environnements, modélisé à partir de la notion d'intervalle de validité temporelle, ne permet pas de déterminer de façon statique, le profil d'ordonnancement associé à une spécification RdPHFT. Aussi le profil d'ordonnancement associé à un RdPHFT ne peut-il être généré dans un système faiblement synchrone qu'à partir de techniques d'ordonnancement dynamiques. Par contre, si l'on se place dans un système synchrone et si l'on interprète un intervalle de validité temporelle comme définissant non plus l'indéterminisme temporel d'un traitement mais sa laxité temporelle, le modèle RdPHFT permet d'envisager des algorithmes d'ordonnancement susceptibles d'exploiter de façon statique la souplesse d'ordonnancement introduite par la laxité temporelle des traitements. Ainsi le développement d'exécutifs d'ordonnancement à partir du modèle RdPHFT doit s'accompagner d'une réflexion sur les implications du modèle en matière de théorie de l'ordonnancement. La deuxième solution consiste à traduire dans une première étape une spécification RdPHFT interprétée vers une architecture de document normalisée de type HTML, MHEG ou Hytime. Le caractère formel du modèle RdPHFT ainsi que le principe de plus petit dénominateur commun hérité du modèle Dexter, facilitent la traduction automatique d'une spécification RdPHFT vers une structure de document hypermédia. Suite à cette opération de traduction d'une spécification RdPHFT ayant suivi tout le cycle de vie précédemment proposé, la structure de document ainsi générée peut être enrichie puis exploitée à partir d'un exécutif HTML, MHEG ou Hytime. Cette deuxième approche offre par rapport à la première les avantages de la normalisation. Cependant les limitations éventuelles des normes existantes au regard des possibilités offertes par le modèle RdPHFT, notamment en matière de sémantique de synchronisation hypermédia, peuvent conduire à utiliser de façon restreinte le

pouvoir d'expression du modèle RdPHFT. Ces deux approches sont en cours d'étude à travers deux thèmes de recherche. Le premier thème s'attache à intégrer des scénarios de synchronisation multimédias dans le World-Wide-Web [CSD95]. La gestion d'objets multimédias dans le World-Wide-Web se fait en introduisant un nouveau type d'objet décrit par une spécification RdPFT et en développant un visualiseur permettant de présenter ces nouveaux objets multimédias. Ce visualiseur multimédia en cours de développement n'est rien d'autre qu'un exécutif de synchronisation susceptible d'interpréter une spécification RdPFT étendue par l'intermédiaire d'attributs HTML [CSD95]. Le deuxième thème de recherche aborde quant à lui la traduction automatique de spécifications RdPHFT vers des structures de documents HTML et MHEG [FCSD96][WSS-SD96]. Plus particulièrement les bases théoriques des techniques de traduction automatiques vers MHEG sont établies par [WSS-SD96] et un traducteur est en cours de développement.

Outre les axes prospectifs précédemment évoqués, d'autres domaines mériteraient d'être étudiés à l'aide du modèle RdPHFT, notamment celui des systèmes coopératifs. Cependant il apparaît dès à présent, à la lumière des contributions apportées par le modèle RdPHFT, que ce modèle propose à ce jour l'une des techniques de spécification les plus avancées pour la modélisation d'applications multimédias/hypermédias, et plus généralement d'applications «à contraintes temps-réel ».



Annexe : Notations

Les notations particulières suivantes sont utilisées dans le cadre de cette thèse:

- \mathbb{Q} , \mathbb{Q}^+ et \mathbb{Q}^- désignent respectivement l'ensemble des nombres rationnels, l'ensemble des rationnels positifs ou nuls et l'ensemble des rationnels négatifs ou nuls.
- $R=(P,T,Pré,Post, M_0,IVT,Syn,AM)$ désigne un RdPFT
- $A = \{(p,t) \in P \times T / (Pré(p,t) \neq 0) \vee (Post(p,t) \neq 0)\}$ désigne l'ensemble des arcs d'un RdPFT R ,
- $A^+ = \{(p,t) \in A / Pré(p,t) \neq 0\}$ désigne l'ensemble des arcs entrant sur les transitions d'un RdPFT,
- $A_k^+ = \{(p_j, t_k) \in A^+\}$ désigne l'ensemble des arcs entrant associés à la transition t_k ,
- $A^- = \{(p,t) \in A / Post(p,t) \neq 0\}$ désigne l'ensemble des arcs sortant des transitions d'un RdPFT,
- $A_k^- = \{(p_j, t_k) \in A^-\}$ désigne l'ensemble des arcs sortant de la transition t_k ,
- $I_k = \{i / a_i \in A_k\}$ désigne l'ensemble des indices des arcs associés à la transition t_k ,
- H désigne une horloge associée à un RdPFT R ,
- $S=(M,I)$ désigne l'état courant d'un RdPFT noté R ,

- $A^s = \left\{ a_{ij} = (p_i, t_j) \in A^+ \mid M(p_i) \geq \Pr \{ p_i, t_j \} \right\}$ désigne l'ensemble des arcs sensibilisés dans l'état courant,
- IDVT est une fonction retournant l'intervalle dynamique de validité temporelle de tout arcs sensibilisé. Ainsi IDVT est définie de la façon suivante:

$$IDVT: A^s \rightarrow I$$

$$\forall a_i \in A^s, IDVT(a_i) = (\alpha_i, d_i, \beta_i) \in I$$

Bibliographie de l'Auteur

- [BCDSS-S94] V. Baudin, G. Ciccelero, M. Diaz, P. Sénac, P. de Saqui-Sannes, *Modélisation et Implantation d'Applications Multimédias*, Rapport LAAS N° 94-329, Contrat Région Midi-Pyrénées N°9300088, 48 p., Septembre 1994.
- [BWSD93] V. Baudin, R. Willrich, P. Sénac, M. Diaz, *Spécification/conception d'une chaîne d'outils pour la production de documents multimédias*, Rapport LAAS N°93454, Contrat CNET France Télécom N°92.1B.178. Lot 4, 51 p., Décembre 1993.
- [CDCS96] J. P. Courtiat, M. Diaz, R. Cruz, P. Sénac, *Formal Models for the Description of Timed Behaviors of Multimedia and Hypermedia Distributed Systems*, invited paper in Computer Communications, 1996.
- [CFLOS-SSW94] J.-P. Courtiat, M. Filali-Amine, A. Leger, R.C. de Oliveira, P. de Saqui-Sannes, P. Sénac, R. Willrich, *Modèles pour la Synchronisation Multimédia et Relations entre Modèles*, Rapport LAAS N° 94-479, Contrat CNET France Télécom N°92.1B.178, projet CESAME, Lot 2, 32 p., Décembre 1994.
- [CSD95] E. Chaput, P. Sénac, M. Diaz, *Introducing Multimedia Synchronization into the World-Wide Web*, IEEE Conference on Multimedia Networking, Aizu, Japan, Oct. 1995.
- [DS92] M. Diaz, P. Sénac, *Réseaux locaux haute vitesse pour le transport d'informations complexes multimodales*, Rapport LAAS N°92291, Contrat Région Midi-Pyrénées N° RECH 9007832, 15 p, Août 1992.
- [DS-SS93] M. Diaz, P. de Saqui-Sannes, P. Sénac, *Un Modèle Formel pour la Spécification de la Synchronisation Multimédia en Environnement Distribué*, Proceedings of CFIP'93, Hermes Editor, Montréal, September 1993.
- [DS93a] M. Diaz, P. Sénac, *Réseaux locaux haute vitesse pour le transport d'informations multimédia*, Rapport LAAS N°92271, Contrat MRT-Etat-Région N°91R07, 30 p, Juin 1993,.
- [DS93b] M. Diaz, P. Sénac, *Time Stream Petri Nets, a Model for Multimedia Streams Synchronization*, Proceedings of MMM'93, MultiMedia Modeling, World Scientific Editor, Singapore, November 1993.
- [DS95] M. Diaz, P. Sénac, *Tutorial on the Modeling of Multimedia Objects and Protocols*, MMM'95, Singapour, November 1995.

- [DVDS92] J.-Y. Dauneau, P. Varaiya, M. Diaz, P. Sénac, *Multimedia: Data Structures and Documents*, Technical Report N° UCB/ERL M92/31, University of California at Berkeley, March 1992.
- [FSS-S95] F. Fabre, P. Sénac, P. de Saqui-Sannes, *A Toolkit for the Modeling of Multimedia Synchronization Scenarios*, Published in the proceedings of MultiMedia Modeling 95, World Scientific Editor, Singapour, Nov. 1995.
- [FSS-S96] F. Fabre, P. Sénac, P. de Saqui-Sannes, M. Diaz, *Un Modèle Formel Unificateur pour les Systèmes Temps-Réels*, published in the Proceedings of Real Time Systems 96, Paris, January 1996.
- [FSS-SD95] F. Fabre, P. Sénac, P. de Saqui Sannes, M. Diaz, *Un modèle et un atelier pour la spécification formelle de systèmes hypermédias*, Rapport LAAS N°95496, Contrat Région Midi-Pyrénées N° RECH/9407495, 10p., Décembre 1995.
- [FUFS-SS93] D. Fazio, A. Ulloa, F. Francés, P. de Saqui-Sannes, P. Sénac, *State of the Art of Multimedia Technology*, COMETT project 92A/1/7131/Cb, Contract report, July 1993.
- [ODS95] P. Owezarski, M. Diaz, P. Sénac, *Modélisation et implémentation de mécanismes de synchronisation multimédia dans une application de visioconférence*, published in Proceedings of CFIP'95, Hermes Editor, 1995.
- [SDS-S93] P. Sénac, M. Diaz, P. de Saqui-Sannes, *A Formal Environment for the Specification and Design of Multimedia Synchronization Scenarios*, short Proceedings of the 4th International Workshop on Network and Operating Support for Digital Audio and Video, Lancaster, November 1993.
- [SDS-S94a] P. Sénac, M. Diaz, P. de Saqui-Sannes, *Un Modèle Formel pour la Spécification de la Synchronisation Multimédia en Environnement Distribué*, Proceedings of Real Time Systems 94, Paris, January 1994.
- [SDS-S94b] P. Sénac, M. Diaz, P. de Saqui-Sannes, *Toward a Formal Specification of Multimedia Synchronization*, invited paper published in "Annals of Telecommunications", May/June 1994.
- [SD94] P. Sénac, M. Diaz, *Time Streams Petri Nets, a Model for Timed Multimedia Informations*, Published in Application and Theory of Petri Nets (IEEE), Lecture Notes in Computer Science, Springer Verlag Editor, 1994.
- [SFCD95] P. Sénac, F. Fabre, E. Chaput, M. Diaz, *Towards a Software Engineering Discipline for the Modeling and Design of Hypermedia Distributed Applications*, to be published in the proceedings of ED-MEDIA' 96, World Conference on Educational Multimedia and Hypermedia 1996, Boston, USA, June 1995.
- [SWD95] P. Sénac, R. Willrich, M. Diaz, *Hypermedia Synchronization Modeling: A Case Study*, Proceedings of the World Conference on Educational Multimedia and Hypermedia, Graz (Austria), June 1995.
- [SS-SW95] P. Sénac, P. de Saqui-Sannes, R. Willrich, *Hierarchical Time Stream Petri Net: A Model for Hypermedia Systems*, Published in Application and Theory of Petri Nets (IEEE), Lecture Notes in Computer Science, Springer Verlag Editor, 1995.

- [SFCD95] P. Sénac, F. Fabre, E. Chaput, M. Diaz, *A Model and Toolkit for the Formal Specification of Weakly Synchronous Systems*, Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Vancouver, Canada, October 1995
- [SDL94] P. Sénac, M. Diaz, A. Léger, *Toward hypermedia synchronization modeling*, Proceedings of the workshop on multipoint communication services, ACM Multimedia 94, San Francisco (USA), May 1994.
- [SDLS-S96] P. Sénac, M. Diaz, A. Léger, P. de Saqui-Sannes, *Modeling Logical and Temporal Synchronization in Hypermedia Systems*, to be published in the IEEE Journal on Selected Areas on Telecommunication, January 1996.
- [SDS-S93] P. Sénac, M. Diaz, P. de Saqui-Sannes, *Synchronisation et Objets Multimédias*, Rapport LAAS N° 92 1B 178, Contrat CNET France Télécom 92.1B.178. Lot 2., 63 p., Juillet 1993.
- [SFCD96] P. Sénac, F. Fabre, E. Chaput, M. Diaz, *Toward a Software Engineering Discipline for the Modeling and Design of Hypermedia Distributed Applications*, to be published in the Proceedings of the World Conference on Educational Multimedia and Hypermedia, Boston, June 1996.
- [WSD94] R. Willrich, P. Sénac, M. Diaz, *Une approche formelle pour la spécification et la conception de documents hypermédias*, Rapport LAAS N°94481, Contrat CNET France Télécom N°92.1B.178, projet CESAME, Lot 2, 68 p., Décembre 1994.
- [WSDS-S96] R. Willrich, P. Sénac, M. Diaz, P. de Saqui-Sannes, *A Formal Framework for the Specification, Analysis and Generation of Standardized Hypermedia Documents*, to be published in the Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Japan, June 1996.
- [WSDS-S96] R. Willrich, P. Sénac, P. de Saqui-Sannes, M. Diaz, *Towards Hypermedia Documents Design*, to be published in the Proceedings of the XIV Brazilian Symposium on Computer Networks, Fortaleza, Brasil May 1996.
- [WS-SSD96] R. Willrich, P. de Saqui-Sannes, P. Sénac, M. Diaz, *Une Approche Formelle pour le Développement de Documents Hypermédias Normalisés*, to be published in the Proceedings of CFIP'96, Editions Hermès, Maroc, Oct. 1996.
- [YSS-S94] J. C. Yee, P. Sénac, P. de Saqui-Sannes, *Ressource Synchronization Specification and Modeling based on Timed Stream Petri Net*, Proceedings of ISMM'94, Hawaii, August 1994.

Bibliographie

- [ACCCD94] P.D. Amer, C. Chassot, C. Connoly, P. Conrad, M. Diaz, *Partial Order Transport Service for Multimedia and other Applications*, IEEE/ACM Transactions on Networking, Vol. 2, No. 5, October 1994.
- [AK91] B. W. Abeysundara, A. E. Kamal, *High-Speed Local Area Networks and their Performance: A Survey*, ACM Computing Surveys, Vol. 23, No. 2, June 1991.
- [AH91] D. P. Anderson and George Homsy, *A Continuous Media I/O Server and Its Synchronization Mechanism*, IEEE Computer, October 1991.
- [AKRS94] A. M. Aras, J. F. Kurose, D. Reeves, H. Schulzrinne, *Real-Time Communication in Packet-Switched Networks*, Proceedings of the IEEE, Vol. 82, No. 1, January 1994.
- [All83] J. F. Allen, *Maintaining Knowledge about Temporal Intervals*, Communication of the ACM, 26, (11), November 1983.
- [Arn92] A. Arnold, *Systèmes de Transition Finis et Sémantique des Processus Communicants*, Masson, Collection Etudes et Recherches en Informatique, 1992.
- [Arn93] A. Arnold, I. Guessarian, *Mathématiques pour l'Informatique*, Masson, Collection Logique Mathématiques Informatique, 1993.
- [App93] W. Appelt, *HyperODA- Extensions for Temporal Relationships*, Proposed Draft Amendment (version 2) ISO/IEC JTC 1/SC 18/WG 3/N 2516, January 1993.
- [App93] W. Appelt, *HyperODA- Extensions for Temporal Relationships*, Proposed Draft Amendment (version 2) ISO/IEC JTC 1/SC 18/WG 3/N 2516, January 1993.
- [App191] Apple Computer Inc., *QuickTime Developer's Guide*, Developer Technical Publications, 1991.
- [BM93] O. Babaoglu and K. Marzullo, *Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms*, Distributed Systems Edited by Sape Mullender, ACM Press, Addison-Wesley, 1993.

- [Bac86] M. J. Bach, *The Design of the Unix Operating System*, Prentice-Hall International, 1986.
- [BAB93] H. Boucheneb, U. Alger and G. Berthelot, *Towards a Simplified Building of Time Petri Nets Reachability Graphs*, Proceedings of the 5th International Workshop on Petri Nets and Performance Models, Toulouse, France, October 1993.
- [BB91] U. Bormann, and C. Bormann, *Standards for Open Document Processing: Current State and Future Developments*, Computer Networks and ISDN Systems, vol. 21, 1991.
- [BCOQ92] F. Baccelli, G. Cohen, G. Jan Olsder, J.P. Quadrat, *Synchronization and Linearity*, John Wiley & Sons, 1992.
- [BD91] B. Berthomieu, M. Diaz, *Modeling and Verification of Time Dependent Systems using Time Petri Nets*, IEEE Transactions on Software Engineering, V 17, N3, March 1991.
- [BH95a] J. P. Bowen, M. G. Hinchey, *Ten Commandments of Formal Methods*, IEEE Computer, April 1995.
- [BH95b] J. P. Bowen, M. G. Hinchey, *Seven More Myths of Formal Methods*, IEEE Software, July 1995.
- [BJ87] K. P. Birman, T. Joseph, *Exploiting Virtual Synchrony in Distributed Systems*, Proc. Eleventh Symp. on Operating Systems Principles ACM, November 1987.
- [Bol90] T. Bolognesi, *From Timed Petri Nets to Timed LOTOS*, IFIP Int Conf on Protocol Specification, Testing and Verification, Ottawa, June 1990, North-Holland.
- [Bra83] G. W. Brams, *Réseaux de Petri: Theorie et Pratique*, Editions Masson, 1983.
- [BS96] G. Blakowski, R. Steinmetz, *A Media Synchronization Survey : Référence Model, Specification, and Case Studies*, IEEE Journal on Selected Areas in Communication, January 1996.
- [Buc95] M. C. Buchanan, *Perspectives on Multimedia Synchronization: Giving Control to Authors*, First International Workshop on Multimedia Synchronization, Boston, 1995.
- [Buf94] J. F. K. Buford, *Architecture and Issues for Distributed Multimedia Systems*, Multimedia Systems, John F. Koegel Buford Contributing Editor, ACM Press, Addison-Wesley, 1994.
- [Bul93] D. C. A. Bulterman, *Specification and Support of Adaptable Networked Multimedia*, ACM Multimedia Systems, Vol 1, No. 2, 1993, p. 68-76.
- [BvL91] D. C. Bulterman, R. van Liere, *Multimedia Synchronization and Unix*, 1991.
- [BvRvL91] D. C. Bulterman, G. van Rossum, R. van Liere, *A Structure for Transportable, Dynamic Multimedia Documents*, Proceedings of USENIX, Nashville, TN, Summer' 1991.
- [BZ93a] M. C. Buchanan and P. T. Zellweger, *Specifying Temporal Behaviour in Hypermedia Documents*, Proceedings of ACM ECHT, Milano, November 1993.

- [BZ93b] M. C. Buchanan and P. T. Zellweger, *Automatically Generating Consistent Schedules for Multimedia Applications*, Multimedia Systems, vol 1 no 2, pp. 55-67, 1993.
- [CCGH92] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, *A continuous media transport and orchestration service*, SIGCOMM'92, Maryland, USA, 1992.
- [CCO96] JP. Courtiat, LFRC. Carmo, RC. de Oliveira *A general-purpose multimedia synchronization mechanism based on causal relations*, IEEE, JSAC 14(1), January 1996.
- [CO95] JP. Courtiat, RC. de Oliveira, *On RT-LOTOS and its application to the formal design of multimedia protocols*, Annals of Telecommunications, 50 (11-12): 888-906, November 1995.
- [COA96] JP. Courtiat, RC. de Oliveira, L. Andriantsiferana, *Formal modeling and verification of multimedia documents*, 1996 Pacific Workshop on Distributed Multimedia Systems, Hong-Kong, June 25-28, 1996.
- [CH74] R. H. Campbell, A. N. Habermann, *The Specification of Process Synchronization by Path Expressions*, In G. Goos and J. Hartmanis, editors, Lectures Notes in Computer Science, No. 16, Operating Systems, p. 89-102, Springer Verlag, 1974.
- [CMQV89] G. Cohen, P. Moller, J. P. Quadrat, M. Viot, *Algebraic Tools for Performance Evaluation of Discrete Event Systems*, Proceedings of the IEEE, vol. 77, no. 1, January 1989.
- [Con87] J. Conklin, *Hypertext: An Introduction and Survey*, IEEE Computer, Vol 20, No. 9, September 1987
- [CBDW92] G. Coulson, G.S. Blair, N. Davies and N. Williams, *Extensions to ANSA for Multimedia Computing*, Computer Networks and ISDN systems, 25, 1992.
- [CBR94] G. Coulson, G. S. Blair, P. Robin, *Micro-kernel support for continuous media in distributed systems*, Computer Networks and ISDN systems, 26, 1994.
- [CdO94] J.P. Courtiat and R. C. de Oliveira, *About time non-determinism and exception handling in a temporal extension of LOTOS*, Proceedings of PSTV '94, Chapman & Hall editor, 1994.
- [Cri89] F. Cristian, *Probabilistic clock synchronization*, Distributed Computing, vol 3, pp. 146-158, Sept. 1989.
- [Dan92] A. Danthine, *Esprit Project OSI 95 New transport services for high-speed networking*, Computer Networks and ISDN Systems 25, 1992.
- [DB93] Andrew W. Davis, Joe Burke, *The Mac Goes to the Movies*, Byte, February 1993.
- [DD94] S. J. DeRose, D. G. Durand, *Making Hypermedia Work, A User Guide to Hytime*, Kuwer Academic Publishers, 1994.
- [DHHKSV94] L. Delgrossi, C. Halstrick, D. Hehmann, RG. Herrtwich, O. Krone, J. Sandvoss and C. Vogt, *Media scaling in a multimedia communication system*, Multimedia Systems, vol. 2 , no. 4, 1994.
- [Dia82] M. Diaz, *Modeling and Analysis of Communication and Cooperation Protocols using Petri Net based Models*, Computer Networks, December 1982.

- [DJ92] B. T. Doshi, P. K. Johri, *Communication protocols for high speed packet networks*, Computer Networks and ISDN Systems 24, 1992.
- [DLCA94] M. Diaz, A. Lozes, C. Chassot, P. Amer, *Partial Order Connections: a new concept for high speed and multimedia services and protocol*, Annals of Telecommunications, May-June 1994.
- [DNNRA93] R. B. Dannenberg, T. Neuendorffer, M. Newcomer, D. Rubine, and D. B. Anderson, *Tactus: Toolkit-level Support for Synchronized Interactive Multimedia*, Multimedia Systems, vol 1, pp. 77-86, 1993.
- [DP94] M. Diaz, G. Pays, *The Cesame Project : Formal Design of High Speed Multimedia Cooperative Systems*, Annals of Telecommunications, May-June 1994.
- [DTH92] S. Dupuy, W. Tawbi, E. Horlait, *Protocols for high-speed multimedia communication networks*, Computer Communications, vol 15, no 6, July/August 1992.
- [DV89] M. Diaz, Ch Vissers, « *SEDOS: Designing Open Distributed Software* », IEEE Software, Vol. 6, N° 6, November 1989.
- [Eri94] H. Erikson, *The multicast backbone*, Communication of ACM, vol 37, no 8, August 1994.
- [EDP92] J. Escobar, D. Deutsch, C. Partridge, *Flow synchronization protocol*, IEEE Globecom Conference, Orlando, Floride, 1992
- [Fer92] D. Ferrari, *Delay jitter control scheme for packet-switching internetworks*, Computer Communications, vol 15, no 6, July/August 1992.
- [Fer94] D. Ferrari, *Network support for multimedia, A discussion of the Tenet Approach*, Computer Networks and ISDN 26, 1994.
- [Fil93] M. Filali, *Une Approche Formelle pour la Description du Comportement Temporel des Applications Multimédia*, Proceedings of CFIP'93, Montréal, September 1993.
- [Fil94] M. Filali, *The Specification of the Temporal Behaviour for Multimedia Scheduling*, Annals of Telecommunications, vol. 49, n° 5-6, Mai-Juin 1994.
- [Fox89] E. A. Fox, *The Coming Revolution in Interactive Digital Video*, Communications of the ACM, Vol 32, N° 7, July 1989.
- [Fre89] K. A. Frenkel, *The Next Generation of Interactive Technologies*, Communications of the ACM, Vol 32, N° 7, July 1989.
- [FSMNM93] K. Fujikawa, S. Shimojo, T. Matsuura, S. Nishio, H. Miyahara, *The Synchronization Mechanisms of Multimedia Information in the Distributed Hypermedia System « Harmony »*, Proceedings of Multimedia Modeling 93, Singapore, Novembre 1993.
- [Gov92] R. Govindan, *Operating Systems Mechanisms for Continuous Media*, Report No. UCB/CSD 92/697, Computer Science Division (EECS) University of California, July 1992.

- [Gre92] J. L. Green, *The Evolution of the DVI System Software*, Communications of the ACM, Vol 35, N° 1, Janvier 1992.
- [Hai95] M. Haindl, *Multimedia Synchronization*, Technical report CS-R9538 1995, CWI Amsterdam, 1995.
- [Hal90] A. Hall, *Seven Myths of formal methods*, IEEE Software, 7 (5),1990.
- [Hal94a] T. R. Halfill, *AMD vs. Superman*, Byte, Vol 19, N° 11, November 1994.
- [Hal94b] T. R. Halfill, *T5: Brute Force*, Byte, Vol 19, N° 11, November 1994.
- [Ham72] C. L. Hamblin, *Instants and Intervals*, Proc. 1st Int. Conf. for the study of time, Springer Verlag, J. T. Fraser at al Editors, 1972.
- [HBvR94] L. Hardman, D. C.A. Bulterman, G. van Rossum, *The Amsterdam Hypermedia Model: Adding tyme to the Dexter Model*, Communications of ACM, Vol 37, No. 2, February 1994 , pp. 50-62.
- [HS91] C. Huitema, J. M. Smith, *Minimal complexity for the simplest protocol*, IEEE Infocom, 1991.
- [HS94] F. Halasz, M. Schwartz, *The Dexter Hypertext Reference Model*, Communications of ACM, Vol 37, No. 2, February 1994 , pp. 30-39.
- [HSA89] M. E. Hodges, R. M. Sasnett, M. S. Ackerman, *A Construction Set for Multimedia Applications*, IEEE Software, Janvier 1989.
- [HSS92] D. B. Hehmann, M. G. Salmony, H. J. Stuttgen, *Transport services for multimedia applications on broadband Networks*, Proc. 1st Int. Conf. for the study of time, Springer Verlag, J. T. Fraser at al Editors, 1992.
- [Hyt92] HyTime Information Technology, *Hypermedia/Time-based Structuring Language (Hytme)*, ISO/IEC DIS 10744, 8, 1992.
- [JSS92] K. Jeffay, D. L. Stone, F. D. Smith, *Kernel Support for Live Digital Audio and Video*, Computer Communication, vol 15 no 6, Aug. 1992.
- [JSS94] K. Jeffay, D. L. Stone, F. D. Smith, *Transport and display mechanisms for multimedia conferencing across packet-switched networks*, Computer Networks and ISDN Systems 26, 1994.
- [Jen94] E. D. Jensen, *A New Generation Open Family of Scaleable Real-Time Computer Systems*, Proc. of RTS'94, Paris, January 1994.
- [JLL77] N. D. Jones, L. H. Landweber, Y. E. Lien, *Complexity of some Problems in Petri Nets*, Theoret. Comput. Sci., vol. 4, pp. 277-299, 1977.
- [JY91] G. Juanolet, Y. Atamna, *Dealing with Arbitrary Time Distributions with the Stochastic Timed Petri Net Model- Applications to Queuing Systems*, International Workshop on Petri Net and Performance Models, Melbourne, Australie, December 1991.

- [Kop93] H. Kopetz, *Scheduling*, Distributed Systems, Edited by Sape Mullender, ACM Press, Addison-Wesley, 1993.
- [LeL94] G. Le Lann, *Why should we keep using Precambrian Design Approaches at the Dawn of the third Millenium*, Proc. of RTS'94, Paris, January 1994.
- [Lév93] P. Lévy, *Les Technologies de l'Intelligence*, Points Sciences, Editions la Découverte, 1993.
- [LG91a] T. D. C. Little, A. Ghafor, *Multimedia Synchronization*, IEEE Data Engineering, September 1991.
- [LG91b] T. D. C. Little, A. Ghafor, *Multimedia synchronization protocols for broadband integrated services*, IEEE Journal on Selected Areas in Communications, 9, (9), December 1991.
- [LG90a] T. D. C. Little, A. Ghafor, *Synchronization and Storage Models for Multimedia Objects*, IEEE Journal on Selected Areas in Communications, 8, (3), April 1990.
- [LG90b] T. D. C. Little, A. Ghafor, *Interval-Based Conceptual Models for Time-Dependant Multimedia Data*, IEEE Transactions on Knowledge and Data Engineering, Vol 8, N° 3, August 1990
- [LG90c] T. D. C. Little, A. Ghafor, *Network Considerations for Distributed Multimedia Object Composition and Communication*, IEEE Network Magazine, Novembre 1990.
- [Lie94] G. Liestol, *Aesthetic and Rhetorical Aspects of Linking Video in Hypermedia*, Proceedings of the ACM European Conference on Hypermedia Technology, pp. 217-222, ACM Press, Edimbourg, September 1994.
- [Lit93] T. D. C. Little, *A framework for synchronous delivery of time-dependent multimedia data*, ACM Multimedia Systems, Vol 1, No. 2, 1993.
- [Lit94] T. D. C. Little, *Time-based Media Representation and Delivery*, IEEE Multimedia Systems, John F. Koegel Buford Contributing Editor, ACM Press, N. Y., 1994.
- [LMM92] I. M. Leslie, D. McAuley, S. Mullender, *Pegasus-Operating System Support for Distributed Multimedia Systems*, ACM Operating Systems Review, November 1992.
- [Lu94] G. Lu, *Advances in Digital Image Compression Techniques*, Computer Communication, Vol. 16, No. 4, Avril 1994.
- [MD89] W. E. Mackay, G. Davenport *Virtual Video Editing in Interactive Multimedia Applications*, Communications of the ACM, Vol 32, N° 7, July 1989.
- [Men82] M. Menasche, *Analyse des Réseaux de Petri Temporisés et Application aux Systèmes Distribués*, Thèse de Docteur Ingénieur, Université Paul Sabatier de Toulouse, 1982
- [Mer74] P. Merlin, *A Study of the Recoverability of Computer Systems*, Thesis, Computer Science Dept., University of California, Irvine, 1974.
- [Mey88] B. Meyer, *Object-Oriented Software Construction*, Prentice Hall, N.Y., 1988.

- [MHEG93] *Information Technology- Coded representation of Multimedia and Hypermedia Information Objects* (MHEG) Part 1, Committee Draft ISO/IEC 13522-1, June 1993.
- [MI80] R. C. O. Martins, K. B. Irani, *Simultaneity of Events in Petri Nets*, Proceedings of the 1980 international Conference on Parallel Processing.
- [MLu67] M. Mac Luhan, *La Galaxie Gutenberg*, Mame, 1967.
- [Mur89] T. Murata, *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, Vol 77, No 4, April 1989.
- [Nel67] T. H. Nelson, *Getting It out of our System*, Information Retrieval a Critical Review, G. Schechter, ed., Thompson Books, Wash D.C., 1967.
- [NKN91] S. Newcomb, N. Kipp, V. Newcomb, *The Hytime Hypermedia/time-based document structuring language*, Communication of the ACM, November 1991.
- [NS92] K. Nahrstedt, J. M. Smith, *The integrated media approach to networked multimedia systems*, Technical report, University of Pennsylvania, Philadelphia, February 1992.
- [NW94] A. L. Narasimha Reddy, James C. Wyllie, *I/O Issues in a Multimedia System*, IEEE Computer, March 1994.
- [NY91] J. Nakajima, M. Yazaki, H. Matsumoto, *Multimedia/Realtime Extensions for the Mach Operating System*, USENIX, Nashville TN, Summer'91.
- [OBD95] P. Owezarsky, V. Baudin, M. Diaz, *Multimedia Teleteaching : introduction of Synchronization and Cooperation Mechanisms in Distance Learning*, In Proceedings of ED-MEDIA'95 World Conference on Educational Multimedia and Hypermedia 1995 , Graz, Austria, June 1995.
- [PLL95] M.J. Pérez-Luque, T.D.C. Little, *A Temporal Framework for Multimedia Synchronization*, First International Workshop on Multimedia Synchronization, Boston , 1995.
- [Pet77] J. L. Peterson, *Petri Nets*, ACM Computing Surveys, Vol 9, N° 3, pp. 223-252, September 1977.
- [Pet81] J. L. Peterson, *Petri Nets Theory and the Modeling of Systems*, Prentice Hall 1981.
- [PR93] B. Prabhakaram, S. V. Raghavan, *Synchronization Models for Multimedia Presentation with User Participation*, Proceedings of ACM Multimedia' 93, Anaheim, Calif , pp. 159-166.
- [Ram74] C. Ramchandani, *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, Project MAC, TR 120, MIT, February 1974.
- [Rav74] K. Ravindran, *Real -Time Synchronization of Multimedia Data Streams in High Speed Networks*, Project MAC, TR 120, MIT, February 1974.
- [Ray91] M. Raynal, *La Communication et le Temps dans les réseaux et les Systèmes Répartis*, Editions Eyrolles, 1991.

- [Red94] A. L. Narasimha Reddy, J. C. Wyllie, *I/O Issues in a Multimedia System*, IEEE Computer, March 1994.
- [RS94] K. Ramamitham, T. A. Stankovic, *Scheduling Algorithms and Operating Systems Support for Real-Time Systems*, Proceedings of the IEEE, Vol. 82, No. 1, January 1994.
- [RS-SC92] L. F. Rust da Costa Carmo, P. de Saqui-Sannes, J. P. Courtiat, *Basic Synchronization Concepts in Multimedia Systems*, Proceedings of the 3rd International Workshop on Network and Operating Support for Digital Audio and Video, La Jolla, California, November 1992.
- [RVR94] S. Ramanathan, H. M. Vin, P. Venkat Rangan, *Towards personalized Multimedia Dial-up Services*, Computer Networks and ISDN Systems, No 26, 1994.
- [RW94] C. Ruemmler, J. Wilkes, *An Introduction to Disk Drive Modeling*, IEEE Computer, March 1994.
- [Sch93] F. B. Schneider, *What Good are Models and What Models are Good*, Distributed Systems Edited by Sape Mullender, ACM Press, Addison-Wesley, 1993.
- [SE93] R. Steinmetz, C. Engler, *Human Perception of Media Synchronisation*, Technical Report, 43.9310, IBM European Networking Center.
- [SF90] P. D. Stotts, R. Furuta, *Temporal Hyperprogramming*, Journal of Visual Languages and Computing, Vol 1, p. 237-253, 1990.
- [SF92] R. Steinmetz and J. C. Fritzche, *Abstractions for continuous media programming*, Computer Communications, vol 15, No 6, Aout 1992.
- [SH94] N. Suri, M. M. Hugue, C. Walter, *Synchronization Issues in Real-Time Systems*, Proceedings of the IEEE, Vol. 82, No. 1, January 1994.
- [SHGC92] D. Sheperd, D. Hutchinson, F. Garcia, G. Coulson, *Protocol support for distributed multimedia applications*, Computer Communication, vol 15, no 6, July/August 1992.
- [SHH92] J. B. Stefani, L. Hazard, F. Horn, *A Computational Model for Distributed Multimedia Applications based on a Synchronous Programming Language*, Computer Communications, vol. 15, no 2, March 1992.
- [Shi79] G. Shiu-Fan Ho, *A Systematic Approach for the Design and Analysis of Distributed Computer Systems*, Thèse de Phd. University of California Berkeley, 1979.
- [Sif77] J. Sifakis, *Use of Petri Nets for Performance Evaluation in Measuring Modeling and Evaluating Computer Systems*, North Holland Publ. Co, pp 75-83, 1977.
- [Sif80] J. Sifakis, *Performance Evaluation of Systems Using Nets*, Net Theory and Applications, Lecture Notes in Computer Science N° 84, pp 307-319, Springer Verlag 1980.
- [Sin92] A. Sinha, *Client-Server Computing*, Communications of the ACM, Vol 35, No. 7, July 1992
- [SMPTE86] *SMPTE Recommended Practice: RP 136-1986 Time and Control Codes for 24, 25 or 30 Frame-Per-Second Motion Picture Systems*, SMPTE J, Aout 1986.

- [Som92] I. Sommerville, *Le Génie Logiciel*, Addison Wesley Editor, 1992.
- [SR94] K. G. Shin, P. Ramanathan, *Real-Time Computing: A New Discipline of Computer Science and Engineering*, Proceedings of the IEEE, Vol. 82, No. 1, January 1994.
- [SS90] D. Sheperd, M. Salmony, *Extending OSI to support synchronization required by multimedia application*, Computer Communication, vol 13, no 7, September 1990.
- [SSF93] V. Saparamadu, A. Seneviratne, M. Fry, *A Review of Inter Media Synchronization Schemes*, Proceedings of Multimedia Modeling 93, Singapore, Novembre 1993.
- [Ste94] P. A. Steenkiste, *A systematic approach to host interface design for high speed networks*, IEEE Computer, March 1994.
- [Ste90] R. Steinmetz, *Synchronization Properties in Multimedia*, IEEE Journal on Selected Areas in Communication, April 1990.
- [Sti95] N. Stiel, *Multimédia: la Nouvelle Frontière*, Universalis 1995, Encyclopaedia Universalis, p. 144-149, 1995.
- [Tan92] A. S. Tanenbaum, *Modern Operating Systems*, Prentice Hall International Editions, 1992.
- [Teil55] P. Teilhard de Chardin, *Le phénomène Humain*, Editions du Seuil, 1955.
- [Tok94] H. Tokuda, *Operating System Support for Continuous Media Applications*, IEEE Multimedia Systems, J. F. Koegel Buford Contributing Editor, ACM Press, N. Y., 1994.
- [TR94] T. Thompson, B. Ryan, *PowerPC 620 Soars*, Byte, Vol 19, N° 11, November 1994.
- [Tur93] K. J. Turner, *Using Formal Description Techniques*, John Wiley & Sons, 1993.
- [VSW94] R. J. Vetter, C. Spell, C. Ward, *Mosaic and the World-Wide Web*, IEEE Computer Magazine, October 1994.
- [WR94] T. Wahl, K. Rothermel, *Representing Time in Multimedia Systems*, Proc of the IEEE Int. Conference on Multimedia Computing and Systems, Boston, USA, May 1994.
- [Wal83] B. Walter, *Timed Petri Nets for Modeling and Analyzing Protocols with Time*, Proc of the IFIP Conf on Protocol Specification, Testing and Verification, III, 1983, North Holland, H. Rudin & C. West eds.
- [Way91] P. Wayner, *Inside Quicktime*, Byte, December 1991.
- [Way94] P. Wayner, *SPARC Strikes Back*, Byte, Vol 19, N° 11, November 1994.
- [WB94] N. Willians, G. Blair, *Distributed Multimedia Applications: a review*, Computer Communications, vol. 17, no. 2, February 1994.

-
- [WDG95] R. Weiss, A. Duda, D. K. Gifford, *Composition and Search with Video Algebra*, IEEE Multimedia, vol. 2, no. 1, 1995.
- [Xuo92] N. H. Xuong, *Mathématiques Discrètes et Informatique*, Masson, Collection Logique Mathématiques Informatique, 1992.
- [Yu89] Yu-Chi Ho, *Dynamics of Discrete Events Systems*, Proceedings of the IEEE, January 1989.
- [Zell89] P. T. Zellweger, *Scripted Documents: A Hypermedia Path Mechanism*, Proceedings of Hypertext'89, November 1989.
- [Zub80] W. M. Zuberek, *Timed Petri Net and Preliminary Performance Evaluation*, 7th Annual Symposium on Computer Architecture, La Baule, May 1980.

Contribution à la Modélisation des Systèmes Multimédias et Hypermédias

Résumé:

La complexité temporelle, logique et sémantique des systèmes multimédias et hypermédias rend nécessaire l'utilisation de techniques formelles permettant de spécifier ces systèmes et de vérifier, en préalable à leur conception, leurs propriétés logiques et temporelles. Cette thèse introduit une nouvelle technique de modélisation de systèmes multimédias et hypermédias basée sur le formalisme des réseaux de Pétri étendus par le temps. Ce modèle appelé Réseaux de Petri à Flux Temporels (RdPFT) permet de spécifier l'indéterminisme temporel intrinsèque aux systèmes distribués multimédias en associant un intervalle temporel aux arcs sortant des places d'un réseau de Petri. Les RdPFT introduisent de plus une sémantique formelle de la synchronisation en environnement faiblement synchrone. Cette sémantique formelle, définie à l'aide de 9 types de règles de tir pouvant être sélectivement associées aux transitions d'un RdPFT, permet de maîtriser l'indéterminisme de synchronisation introduit par la variabilité temporelle des traitements. Ainsi, les RdPFT permettent de modéliser de façon complète, précise et aisée les contraintes de synchronisation des systèmes multimédias. Cette thèse considère également la classe des RdPFT construits à l'aide des opérateurs de l'algèbre temporelle de Allen. Cette classe, appelée Réseaux de Petri Structurés à Flux Temporels (RdPSFT), permet une vérification aisée des propriétés temporelles d'un système multimédia et offre de plus des bases formelles pour la représentation abstraite des structures logiques et temporelles d'un système multimédia. Finalement cette thèse introduit une extension des RdPFT, appelée Réseaux de Petri Hiérarchisés à Flux Temporels (RdPHFT). Cette extension, bénéficiant de tous les acquis théoriques des RdPFT, permet de modéliser à l'aide d'un formalisme unificateur les composants d'un système hypermédia et introduit une sémantique formelle de la notion de synchronisation hypermédia. De plus les RdPHFT offrent un support méthodologique permettant une spécification modulaire et hiérarchisée des systèmes hypermédias, et plus généralement des systèmes «à contraintes temporelles».

Mots-clefs : *Multimédia, Hypermédia, Modélisation, Techniques Formelles, Réseaux de Petri, Temps-réel, Asynchronisme, Vérification, Validation.*

Towards Multimedia and Hypermedia Systems Modeling

Summary:

The temporal, logical and semantic complexity of multimedia and hypermedia systems makes necessary the use of formal techniques for their specification and for verifying their logical and temporal properties. This thesis introduces first a new formal technique, based on extended timed Petri Nets, for the modeling of multimedia and hypermedia systems. This model, called Time Stream Petri Nets (TSPN), allows the temporal non-determinism of distributed multimedia systems to be specified by associating arcs outgoing from places with temporal intervals. Moreover TSPN introduce a formal semantics of synchronization in weakly synchronous systems. This formal semantics, defined with the help of 9 firing rules which can be selectively associated to transitions, allows the synchronization non-determinism induced by the temporal variability of processings to be mastered. Therefore TSPN allow an accurate, complete and easy modeling of multimedia synchronization constraints. This thesis consider also the subclass of TSPN build with the help of Allen's synchronization operators. This class, called Structured Time Stream Petri Net (STSPN) allows temporal properties of multimedia systems to be easily checked and offers formal bases for the abstract representation of multimedia logical and temporal structures. Finally this thesis introduces another new model extending TSPN and called Hierarchical TSPN (HTSPN). This extension which inherits TSPN properties allows hypermedia components to be modeled with the help of a unifying formalism and introduces a formal semantics of hypermedia synchronization. Moreover HTSPN offer a modular and hierarchical methodology for the specification and validation of hypermedia systems and more generally of « time constrained » systems.

Keywords : *Multimedia, Hypermedia, Modeling, Formal Techniques, Petri Nets, Real-Time, Asynchronism, Verification, Validation*