

AIDE A LA MODELISATION DES SYSTEMES COOPERATIFS

Nicolas RIVIERE*

Directeurs de thèse : Brigitte PRADIN-CHEZALVIEL et Robert VALETTE

Laboratoire d'accueil :
LAAS-CNRS
7, Avenue du Colonel Roche
F-31077 Toulouse Cedex 4

Etablissement d'inscription :
INSA-Toulouse
135, Avenue de Ranguel
F-31077 Toulouse Cedex 4

Résumé

Le domaine d'application concerne à la fois les systèmes coopératifs et les systèmes multimedia. Nous cherchons à développer des outils d'aide à la mise en oeuvre de ces systèmes à l'aide d'agents. Nous focalisons notre travail sur le respect de contraintes temporelles aussi bien à l'intérieur des agents qu'au niveau de leur coordination. Pour cela nous développons une approche formelle fondée sur les réseaux de Petri et la logique linéaire. Cette dernière nous permet d'obtenir tous les ordres partiels entre les événements cohérents avec l'ensemble des contraintes. Pour un ordre partiel donné nous pouvons définir un ordonnancement des événements vérifiant les contraintes temporelles.

Mots clefs

Logique linéaire, Réseaux de Petri temporels, TSCP, Graphes potentiels événements, Ordres partiels

1 INTRODUCTION

La distribution des acteurs participant à la réalisation d'un but commun les amène à utiliser des méthodes spécifiques à leur coopération. Le domaine d'application concerne à la fois les systèmes coopératifs et les systèmes multimedia. Nous cherchons à développer des outils d'aide à la mise en oeuvre de ces systèmes à l'aide d'agents, qui peuvent être aussi bien des humains que des ressources informatiques (matérielles ou logicielles). Nous focalisons notre travail sur le respect de contraintes temporelles aussi bien à l'intérieur des agents qu'au niveau de leur coordination. Pour cela nous développons une approche formelle fondée sur les réseaux de Petri et la logique linéaire. Cette dernière nous permet d'obtenir tous les ordres partiels entre les événements cohérents avec l'ensemble des contraintes. Pour un ordre partiel donné nous pouvons définir un ordonnancement des événements vérifiant les contraintes temporelles.

La logique linéaire est utilisée pour analyser les modèles afin de trouver tous les ordres partiels (lorsqu'il en existe au moins un) conduisant d'un état à un autre par construction d'arbre de preuves. Son utilisation conjointe avec les réseaux de Petri provient du résultat de base concernant l'équivalence entre le problème de l'accessibilité dans un réseau de Petri et la prouvabilité d'un séquent en logique linéaire [5]. La recherche peut se faire soit vers l'avant soit vers l'arrière [6] [1].

Pour pouvoir prendre en compte l'aspect temporel du travail coopératif, nous utilisons les extensions temporelles des réseaux de Petri pour modéliser nos systèmes. Des travaux sur les modèles t-temporisés [8] et t-temporels [9] nous ont permis de caractériser des durées de scénario avec des dates symboliques et en fonction de la sémantique utilisée de pouvoir invalider des scénarios en raisonnant sur les domaines de valeurs possibles. L'utilisation du modèle p-temporel des réseaux de Petri nous a amenés à réaliser un travail sur la traduction des réseaux de Petri en problèmes de satisfaction de contraintes (CSP) à l'aide de la logique linéaire [7]. Nous allons nous focaliser dans cet article sur cet aspect. Pour un ordre partiel donné, nous allons nous attacher à montrer comment nous pouvons

*nriviere@laas.fr

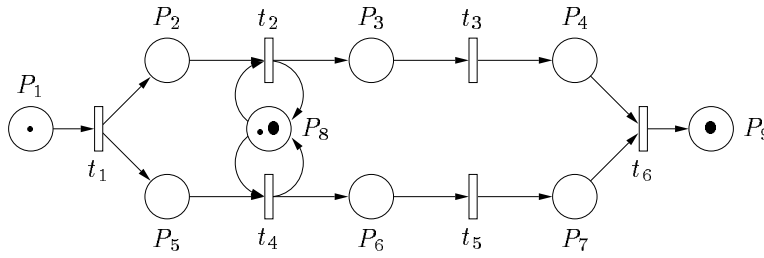


FIG. 1: Réseau de Petri avec conflit

définir un ordonnancement des événements vérifiant les contraintes temporelles imposées par notre système. Pour cela, nous introduisons l'extension temporelle des CSP (TCSP) par l'intermédiaire de leur représentation graphique AOA (graphe d'activités sur les arcs) [10].

2 LOGIQUE LINEAIRE ET ORDRES PARTIELS

2.1 RAPPELS SUR LA LOGIQUE LINEAIRE

Un réseau de Petri définit des relations de précédence entre les transitions. L'analyse de l'accessibilité (entre deux marquages) est basée de façon classique sur des séquences de tir dans lesquelles le franchissement des transitions apparaît dans un ordre total. L'utilisation de la logique linéaire [4] nous permet d'obtenir des ordres partiels cohérents avec les relations de précédence induites par le réseau [3]. Pour cela nous utilisons l'équivalence entre accessibilité dans un réseau de Petri et prouvabilité de séquent en logique linéaire [5] que nous allons présenter en suivant.

Une formule logique est associée à chaque marquage et à chaque instance de franchissement d'une transition. Un marquage M est un monôme en \otimes , *i.e.*, que l'on écrit sous la forme $A_1 \otimes A_2 \otimes \dots \otimes A_k$, où les A_i sont les noms des places du réseau de Petri. Pour toute place A_k qui contient plusieurs jetons, la proposition A_k apparaît autant de fois que la place contient de jetons.

Une transition est une formule du type $M_1 \multimap M_2$ où M_1 et M_2 sont les marquages, soit les fonctions Pre et Post des transitions. Cette expression représente le franchissement d'une transition. Elle apparaîtra autant de fois dans le séquent que cette transition doit être franchie. Sur la figure 1, le franchissement de t_1 sera traduit en logique linéaire par $P_1 \multimap P_2 \otimes P_5$.

Un séquent est associé à un scénario où le marquage initial et les franchissements des transitions sont les prémisses et le marquage final, la conclusion. Nous prouvons alors le séquent à l'aide des règles du calcul des séquents. Le fait que les atomes correspondant aux propositions logiques puissent être comptés, produits et consommés exactement comme des jetons de places de réseau de Petri montre bien que la prouvabilité d'un séquent est équivalente à l'accessibilité du marquage final depuis l'initial. L'ensemble des franchissements de transition montre clairement quelles transitions sont tirées. Durant la preuve, toutes les contraintes générées par le réseau et le marquage initial sont vérifiées.

Le séquent $M, \sigma \vdash M'$ représente un scénario où $\sigma = t_i, \dots, t_n$ est la liste non ordonnée des différentes instances de tir des transitions concernées et M et M' sont les marquages initiaux et finaux.

La preuve est toujours matérialisée par un arbre qui se lit de bas en haut dans le cadre du calcul des séquents. La preuve démarre par le séquent à prouver et se termine lorsque les feuilles de l'arbre sont des séquents *identité*, $A \vdash A$ par exemple. Plusieurs arbres de preuves sont possibles mais la preuve est construite de manière canonique. Dans cet article, nous nous plaçons dans le fragment ILL (Intuitionist Linear Logic) de la logique linéaire ce qui réduit l'utilisation des connecteurs au nombre de deux :

- le connecteur \otimes représente la disponibilité simultanée de ressources,
- le connecteur \multimap est l'implication linéaire qui représente la causalité car les atomes du côté gauche du connecteur doivent être consommés pour produire ceux du côté droit.

2.2 DERIVATION D'UN ORDRE PARTIEL

Le début d'une preuve consiste à éliminer le connecteur \otimes dans le marquage initial. Cela va nous permettre de travailler avec une liste de jetons (atomes logiques) logiquement indépendants mais pas nécessairement présents simultanément plutôt qu'avec des marquages. Durant la preuve, ces jetons correspondent à des relations de précédence. Contrairement aux travaux menés dans [8] et [9], les

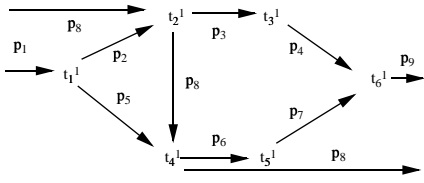


FIG. 2: Ordre partiel 1

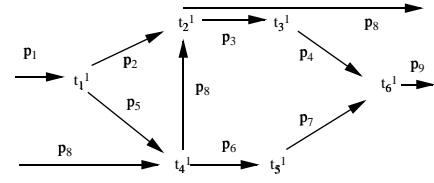


FIG. 3: Ordre partiel 2

atomes de nos arbres de preuve seront estampillés par des instances de tir de transitions et non plus par des dates symboliques de production et de consommation. Soit t_i^j le $j^{\text{ème}}$ tir de la transition t_i . Si un jeton est produit par un tir de transition t_i^j et est consommé par t_k^l , cela signifie que t_i^j doit précéder t_k^l (noté $t_i^j \prec t_k^l$). Une paire (t_i^j, t_k^l) est associée à chaque jeton : ils représentent respectivement le tir de transition qui a produit le jeton et celui qui l'a consommé. Quand un tir n'est pas encore défini (encore inconnu), il est noté par un point. A la fin de la preuve, il suffit juste de collecter tous les atomes qui sont apparus. En effet chacun d'eux correspond à une relation de précédence et l'ensemble de ces derniers définit l'ordre partiel.

Durant la preuve, chaque pas permet l'élimination d'une formule correspondant à une transition. A chaque pas de la preuve, une transition t_i^j est éliminée de la liste des formules, les jetons consommés par t_i^j sont enlevés de l'ensemble des atomes avant franchissement et ceux produits ajoutés. Si à un pas donné de la preuve plus d'une élimination est possible et si les transitions et les jetons concernés sont disjoints, alors la preuve qui en découlera sera la même quel qu'en soit l'ordre d'élimination. Par contre si les transitions ou les jetons sont partagés (conflits jetons ou conflits transitions définis comme dans [8]), alors un arbre de preuve pour chaque ordre partiel devra être dérivé car chaque arbre caractérise un ordre partiel différent.

2.3 EXEMPLE

Considérons le modèle de la figure 1 où il y a deux chemins distincts à exécuter et une ressource (H) partagée. Les événements initiaux et finaux sont t_1^1 et t_6^1 . Les deux chemins contiennent deux actions chacun : t_2^1 et t_3^1 pour le premier et t_4^1 et t_5^1 pour le second. Les actions t_2 et t_4 partagent la même ressource. Le marquage initial est $P_1 \otimes P_8$ et le marquage final est $P_8 \otimes P_9$.

Après l'élimination de la transition t_1^1 , il y a un conflit transition entre t_2 et t_4 car le marquage P_2, P_5, P_8 est atteint et que les deux transitions doivent consommer P_8 . Deux ordres partiels sont ainsi dérivés (figures 2 et 3). P_8 apparaît trois fois dans chaque ordre partiel car il en existe un exemplaire initialement qui est consommé et produit par t_2^1 puis reconsumé et reproduit par t_4^1 .

3 ORDONNANCEMENT TEMPOREL A L'AIDE DE GRAPHES AOA

3.1 INTRODUCTION DES CSP ET DES GRAPHES POTENTIELS EVENEMENTS

Dans un travail précédent sur les relations entre les réseaux de Petri et les graphes de contraintes [7], nous avons vu qu'il était possible de traduire l'ensemble des contraintes générées par un réseau de Petri, un marquage initial et une liste de franchissement de transitions sous la forme d'un CSP (Constraint Satisfaction Problem) [12]. Toutefois, le CSP obtenu est une forme dégénérée avec une contrainte unique qui est la disjonction de l'ensemble des ordres partiels solutions des contraintes.

Un CSP est défini par un triplet (X, D, C) où :

- $X = x_1, \dots, x_n$ est un ensemble de n variables,
- $D = d_1, \dots, d_n$ est l'ensemble des domaines des variables,
- $C = c_1, \dots, c_e$ est un ensemble de e contraintes, où chaque c_i est définie par l'ensemble des variables $X(c_i) \subset X$ impliqué dans la contrainte c_i et par une relation $R(c_i)$ entre les variables de $X(c_i)$.

Le fait de travailler avec des contraintes temporelles nous amène à travailler avec une certaine classe de CSP : les TCSP. Un TCSP (temporal constraints satisfaction problem) est une classe particulière

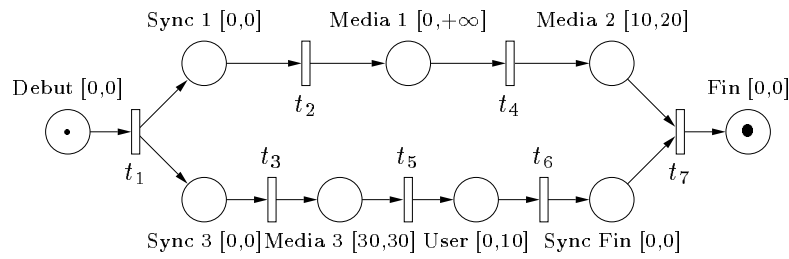


FIG. 4: Réseau de Petri P-temporel

des CSP où l'ensemble X des variables dénote un ensemble d'entités temporelles (dates, intervalles temporels, durées) et les contraintes C représentent les relations logiques ou temporelles possibles entre les variables. Toutes les contraintes de type TCSP sont binaires [11]. Il faut noter que comme nous ne travaillons qu'avec un seul intervalle temporel liant deux variables, nous nous plaçons dans le cadre d'un STP (Simple Temporal Problem). Pour chaque ordre partiel il y a un ensemble de contraintes suffisamment simple pour être résolu en un temps polynômial et pour permettre des procédures simples d'*arc consistence*.

Pour prendre en compte le temps et représenter de manière graphique un TCSP, nous utilisons les graphes AOA (activity-on-arc) [2]. Un graphe AOA est un graphe potentiel événement où les noeuds représentent les variables X du TCSP (soit des événements) et les arêtes les contraintes temporelles entre deux événements. A chaque noeud $i \rightarrow j$ est associée une contrainte binaire c_{ij} . Les noeuds représentent des événements ou des pas relatifs à des débuts ou des fins d'activités. La longueur d'un arc représente la durée de l'activité attachée à cet arc. Etant donné que nous sommes capables de générer un CSP à partir d'un réseau de Petri, nous pouvons donc associer un graphe AOA à un réseau temporel ou temporisé.

3.2 LES RESEAUX DE PETRI ET LE TEMPS

Ordonnancer un ensemble d'opérations est équivalent à trouver un ordre partiel parmi les tirs de transition correspondant aux opérations à faire et à fixer leur date de tir de façon à respecter toutes les contraintes. Dans notre contexte, le modèle réseau de Petri doit être complété par des contraintes temporelles (durées d'opérations, dates voulues, etc...). Il existe des réseaux de Petri, enrichis par des extensions temporelles, qui permettent de spécifier les contraintes complexes impliquant le temps et les ressources. Ce sont les modèles p-temporel, t-temporel, p-temporisé et t-temporisé.

Dans notre approche, nous utilisons le modèle p-temporel car il a été montré dans [7] que lors d'une traduction d'un réseau de Petri temporel en un graphe AOA, il était plus naturel d'associer des durées aux places plutôt qu'aux transitions car les contraintes de précedence sont générées par des places.

Un réseau de Petri p-temporel (figure 4) est un réseau de Petri P avec une fonction associant une durée minimale d_{imin} et une durée maximale d_{imax} à chaque place p_i . Chaque jeton doit rester au moins d_{imin} dans p_i et doit être consommé par le tir d'une transition avant d_{imax} .

Ce modèle associe deux contraintes à chaque relation de précedence, une pour la durée minimale et une pour la durée maximale. Il est possible d'avoir des circuits positifs ce qui signifie que l'ensemble de nos contraintes est incohérent. En effet il n'est pas plausible qu'un circuit composé de durées minimales soit supérieur à un circuit composé de durées maximales. Les réseaux de Petri p-temporel sont capables de prendre en compte les échéances et sont plus généraux que les réseaux p-temporisé. La recherche d'une solution peut être faite en utilisant la programmation linéaire.

3.3 CALCUL DES DOMAINES DE DATES DE TIR : EXEMPLE

Une fois la traduction de l'ordre partiel (obtenu par un arbre de preuve en logique linéaire) et des contraintes temporelles effectuée sous la forme d'un graphe AOA, les techniques classiques de propagation des contraintes permettent d'obtenir les dates de franchissement *au plus tôt* et les dates de franchissement *au plus tard* compatibles avec les contraintes.

Pour illustrer notre méthode, nous allons considérer le scénario multimedia qui décrit la présentation simultanée de deux médias (1 et 2) avec un média interactif 3. La présentation du média 1 a une durée de présentation indéterminée $[0, +\infty[$ en secondes. Le média 2, qui démarre après le média 1, a une durée comprise entre $[10, 20]$. 3 est un média interactif qui dure au maximum 40 secondes. L'aspect interactif vient du fait que l'utilisateur a la possibilité d'interrompre ce média entre 30 et 40

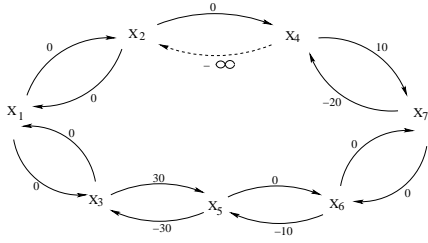


FIG. 5: Graphe AOA initial

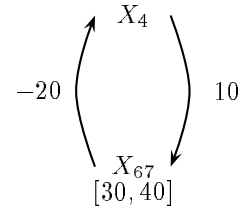


FIG. 6: Graphe AOA réduit

secondes sans quoi sa présentation se termine automatiquement lorsque la durée maximale du média 3 est atteinte. Les présentations des médias 1 et 3 débutent en même temps et celles des médias 2 et 3 doivent se terminer en même temps. D'après ces spécifications nous devons donc trouver une durée maximale de présentation du média 1 afin de pouvoir satisfaire les contraintes de synchronisation de notre scénario.

A partir des spécifications, un modèle réseau de Petri p-temporel est créé (figure 4). Il est à noter que les noms des places de ce réseau ne correspondent pas aux noms des médias spécifiés ci-dessus. La branche composée des places *Sync1*, *Media1* et *Media2* représente les présentations successives des médias 1 et 2. La branche composée des places *Sync3*, *Media3*, *User* et *SyncFin* représente la présentation du média 3 avec l'interaction de l'utilisateur modélisée par la place *SyncFin* : en effet, si la transition t_6 est franchie avant que la durée liée à la place *User* n'atteigne sa valeur maximale (10 secondes), il faut immédiatement sensibiliser la transition t_7 d'où une attente nulle sur *SyncFin*.

En considérant le modèle de la figure 4, nous générons le graphe AOA de la figure 5. Dans notre exemple, étant donné que la durée maximale de *Media1* (égale à $+\infty$) ne contraint pas l'activité liée à la place *Media1*, l'arc inverse (en pointillé sur la figure 5) reliant la variable X_4 à la variable X_2 est éliminé. Il nous reste maintenant à quantifier le domaine de l'intervalle lié à X_4 . Pour cela, on résout un système linéaire d'inéquations déduit du graphe AOA de la figure 5.

Nous énumérons toutes les inéquations du système linéaire :

$$\begin{array}{l|l|l} X_2 - X_1 \geq 0 & X_3 - X_1 \geq 0 & X_6 - X_5 \geq 0 \\ X_1 - X_2 \geq 0 & X_1 - X_3 \geq 0 & X_5 - X_6 \geq -10 \\ X_4 - X_2 \geq 0 & X_5 - X_3 \geq 30 & X_7 - X_6 \geq 0 \\ X_7 - X_4 \geq 10 & X_3 - X_5 \geq -30 & X_6 - X_7 \geq 0 \\ X_4 - X_7 \geq -20 & & \end{array}$$

Cela nous amène au système final :

$$X_{67} - X_4 \geq 10 \quad (1)$$

$$X_4 - X_{67} \geq -20 \quad (2)$$

$$\text{avec } 30 \leq X_{67} \leq 40 \quad (3)$$

D'après ce dernier système nous obtenons le nouveau graphe AOA de la figure 6.

Cela correspond bien aux spécifications de notre document multimedia. En effet l'utilisateur peut stopper la présentation du média 3 entre 30 et 40, ce que montre l'encadrement de X_{67} . Cet encadrement correspond au domaine de valeurs que peut prendre cette variable.

A partir des inéquations (1) et (2) et du domaine incertain de X_{67} , nous avons deux cas possibles de valeurs pour X_4 :

- 1er cas : nous pouvons trouver un domaine de valeurs pour X_4 pour lequel il existe au moins une solution possible pour X_{67} .
- 2ème cas : nous pouvons trouver une valeur pour X_4 pour laquelle toutes les valeurs de X_{67} sont possibles.

En prenant la première inéquation et en posant $X_{67} = 30$, nous nous plaçons dans le cas le plus contraint pour les valeurs possibles de X_4 : nous obtenons $X_4 \leq 20$. En posant $X_{67} = 40$, nous sommes dans le cas le moins contraint et nous obtenons $X_4 \leq 30$.

Maintenant, avec la deuxième inéquation, si nous posons $X_{67} = 40$ nous sommes dans le cas le plus contraint et $X_4 \geq 20$. Et inversement avec $X_{67} = 30$ nous sommes dans le cas le moins contraint, ce qui nous donne $X_4 \geq 10$.

Nous pouvons dire que l'encadrement $20 \leq X_4 \leq 20$, soit $X_4 = 20$, qui correspond au cas le plus contraint, caractérise l'ensemble des valeurs de X_4 pour lequel toutes les valeurs de X_{67} satisfont les contraintes de synchronisation. Par contre l'encadrement $10 \leq X_4 \leq 30$, qui correspond au cas le moins contraint, caractérise l'ensemble des valeurs de X_4 pour lequel il existe au moins une valeur de X_{67} qui satisfait les contraintes de synchronisation du scénario. L'intervalle temporel de la place D du réseau p-temporel de la figure 4 ne doit donc plus être $[0, \infty]$ mais l'intervalle de l'un des deux cas caractérisés ci-dessus suivant la flexibilité que l'on veut laisser à l'utilisateur pour interagir. Pour le plus de flexibilité on choisira $[20, 20]$ sinon on choisira $[10, 30]$ sachant que l'on contraindra l'utilisateur à ne pouvoir éventuellement agir qu'en un instant donné.

4 CONCLUSION

Dans cet article nous avons montré que l'utilisation conjointe des réseaux de Petri p-temporel, des graphes AOA et des TCSP nous permettait de trouver au moins une solution satisfaisante, s'il en existe une, à un problème d'ordonnement en temps contraint pour chaque ordre partiel sans avoir à construire le graphe de tous les états possibles de notre modèle. La solution obtenue n'est bien sûr valable que pour l'ordre partiel généré par l'utilisation de la logique linéaire et dans lequel nous avons propagé les contraintes temporelles. Dans le cas où un conflit apparaît, autant d'ordres partiels que de chemins différents devront être générés pour lesquels nous pouvons trouver une solution qui sera différente pour chaque ordre partiel.

En perspective, il serait intéressant de comparer les solutions obtenues pour chaque ordre partiel avec les solutions obtenues en se basant sur le graphe d'états complet car il n'est pas certain que les solutions trouvées à partir du graphe global soient l'union des solutions trouvées pour chaque ordre partiel. Pour cela, un outil est prévu afin d'automatiser la génération des ordres partiels pour faciliter notre travail, outil basé sur la réécriture des règles de logique linéaire à l'aide d'un langage de programmation. Il serait aussi intéressant de récupérer ces ordres partiels et de les traiter avec un outil qui utiliserait un algorithme optimal de programmation linéaire.

Références

- [1] Hammid DEMMOU, Sarhane KHALFAOUI, RIVIERE Nicolas, Edwige GUILHEM, and Robert VALETTE. Extracting critical scenarios from a petri net model using linear logic. In *Journal Européen des Systèmes Automatisés, JESA'02*, summer 2002. à paraître.
- [2] S.E. Elmaghraby. *Activity networks : project planning and control by network models*. John Wiley & sons, New York, 1977.
- [3] V. Gehlot. *A proof theoretic approach to semantics of concurrency*. Phd thesis, University of Pennsylvania, 1992.
- [4] Jean-Yves GIRARD. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [5] François GIRAULT. *Formalisation en logique linéaire du fonctionnement des réseaux de Petri*. PhD thesis, Université Paul Sabatier, Toulouse, France, Décembre 1997.
- [6] Sarhane KHALFAOUI, Edwige GUILHEM, Hammid DEMMOU, and RIVIERE Nicolas. Extraction des scénarios critiques à partir d'un modèle rdp à l'aide de la logique linéaire. In *Modélisation des systèmes réactifs, MSR'01*, pages 409–424, ENSICA, Toulouse, France, 17-19 octobre 2001.
- [7] C. Mancel, P.Lopez, N.Rivière, and R. Valette. Relationships between petri nets and constraint graphs : application to manufacturing. In *Rapport LAAS 01396*, Toulouse, France, September 2001. à paraître dans IFAC'02.
- [8] Brigitte PRADIN-CHEZALVIEL, Luis Allan KÜNZLE, and Robert VALETTE. Scenario durations characterization of t-timed Petri nets using Linear Logic. *PNPM*, 1999.
- [9] Nicolas RIVIERE, Brigitte PRADIN-CHEZALVIEL, and Robert VALETTE. Reachability and temporal conflicts in t-time Petri nets. *PNPM*, 2001.
- [10] Nicolas RIVIERE, Brigitte PRADIN-CHEZALVIEL, and Robert VALETTE. Propagation de contraintes et ordonnancement de documents multimedia. In *Journées Doctorales Informatique et Réseaux, JDIR'02*, pages 211–218, Toulouse, France, 4-6 mars 2002.
- [11] E. Schwalb and L. Vila. Temporal constraints : a survey. *Constraint : an International journal (Kluwer Academic)*, 2 :129–149, 1998.
- [12] E.P.K. Tsang. *Foundations of constraint satisfaction*. Academic Press Ltd., London, 1993.