
Contribution à la modélisation des systèmes à temps contraint et application aux systèmes multimédias

Thèse de l'université Toulouse III

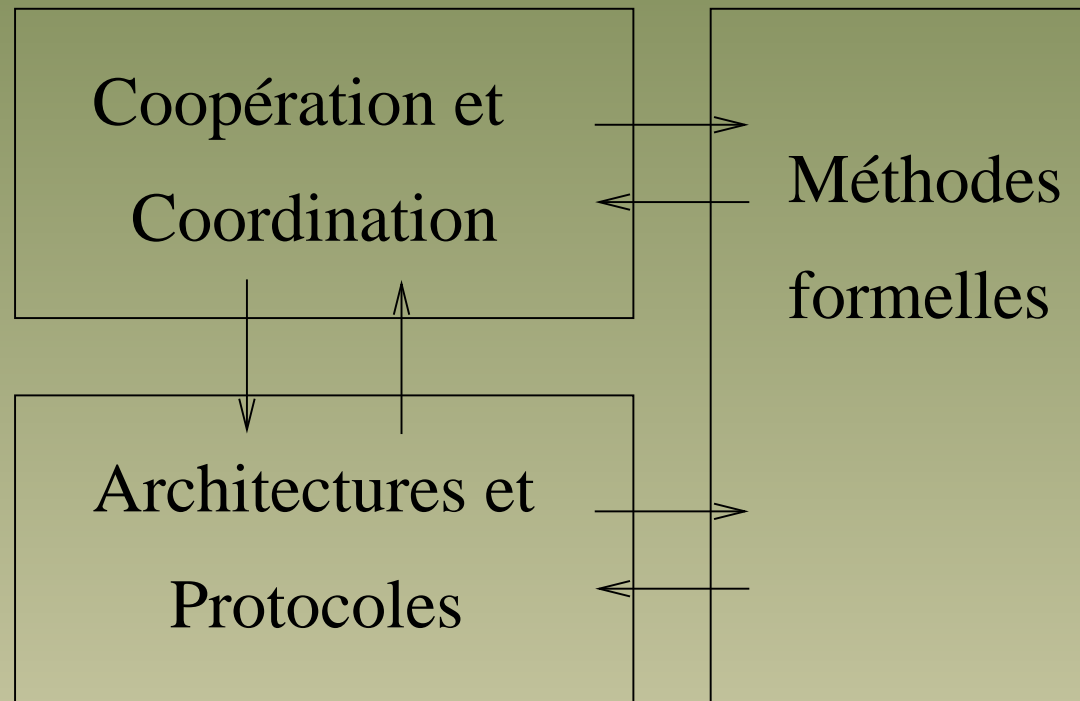
Marc BOYER

sous la direction de Michel DIAZ

LAAS/CNRS

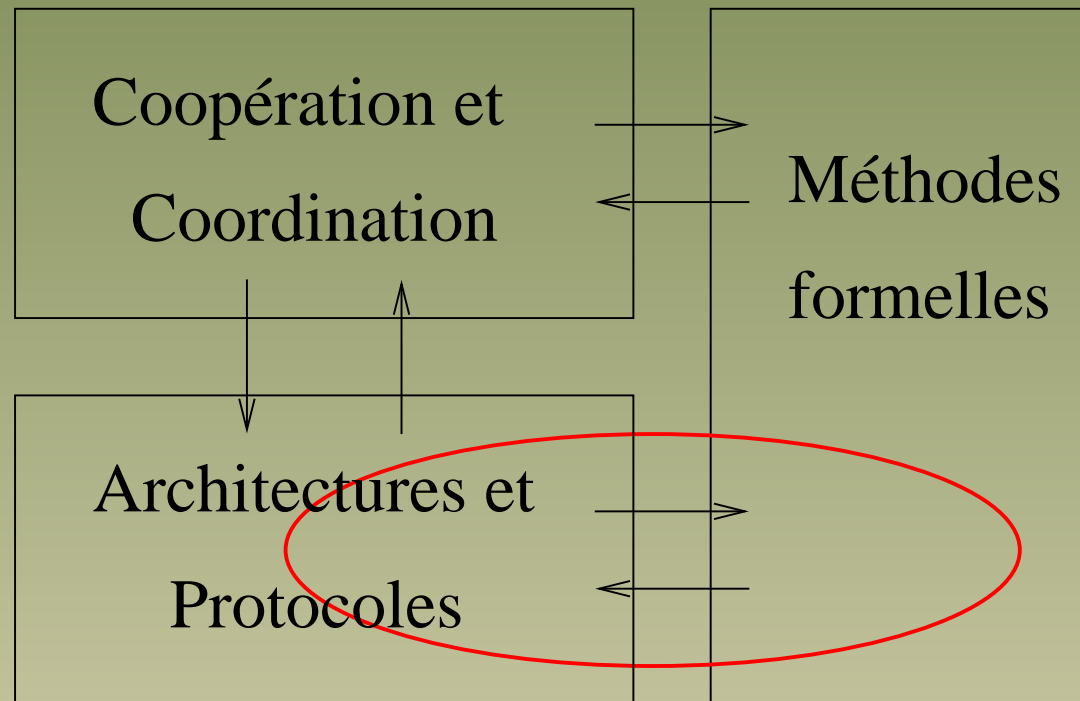
Contexte de recherche

Outils et Logiciels pour la Communication



Contexte de recherche

Outils et Logiciels pour la Communication



Plan de la présentation

Plan de la présentation

1. Modéliser les applications multimédias

Plan de la présentation

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations

Plan de la présentation

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence

Plan de la présentation

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

Problématique personnelle

Applications multimédias distribuées

- multimédia :

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :
 - sans connaissance globale

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :
 - sans connaissance globale
 - communication non fiable

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :
 - sans connaissance globale
 - communication non fiable
 - communication avec giges

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :
 - sans connaissance globale
 - communication non fiable
 - communication avec giges

⇒ besoin d'un modèle et d'une sémantique formelle

Problématique personnelle

Applications multimédias distribuées

- multimédia :
 - traitement temps-réel
 - utilisateur humain
- distribution :
 - sans connaissance globale
 - communication non fiable
 - communication avec gigue

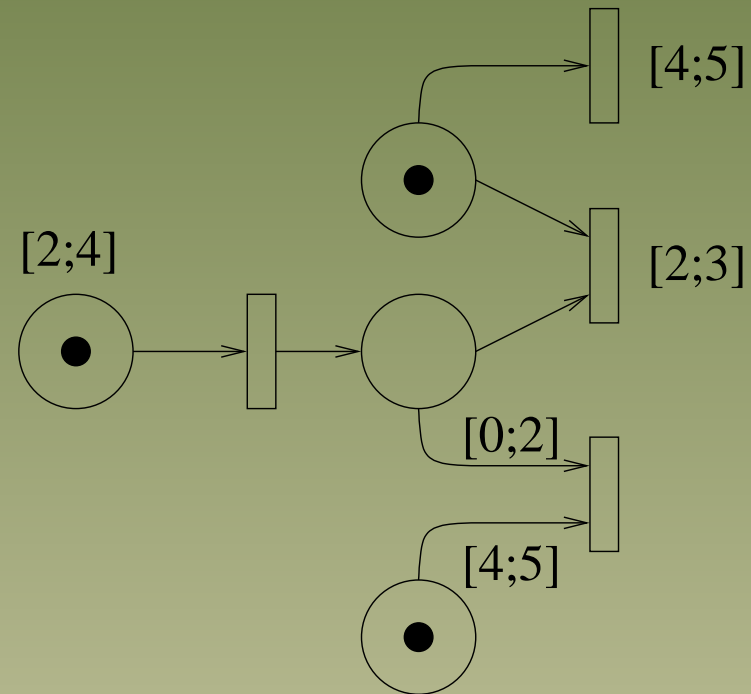
⇒ besoin d'un modèle et d'une sémantique formelle
pour les applications

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art : 2 approches
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(a.1) Réseaux de PETRI à intervalles

Réseaux de PETRI avec intervalles sur les :

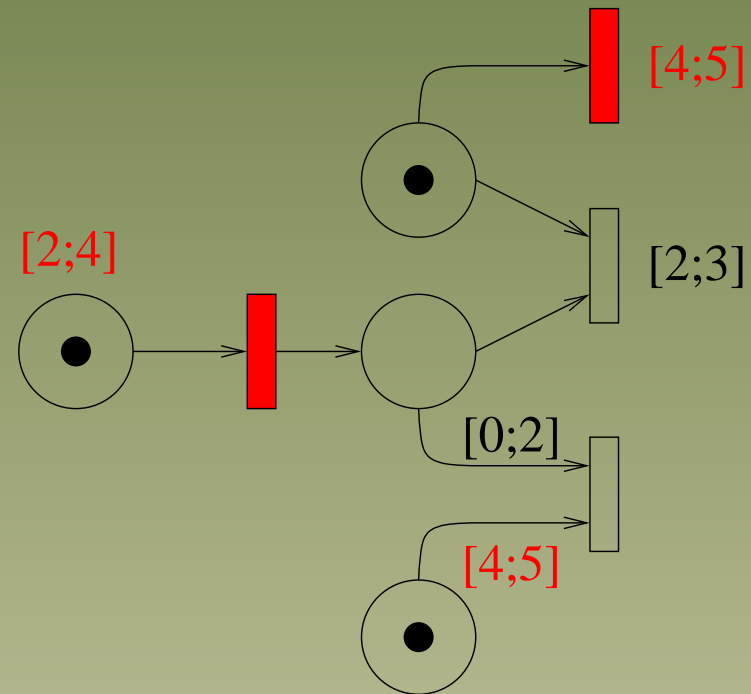
- transitions (t-RdP)
[MERLIN 74]
- places (p-RdP)
[KHANSA 97]
- arcs (a-RdP)
[WALTER 83]



(a.1) Réseaux de PETRI à intervalles

Réseaux de PETRI avec intervalles sur les :

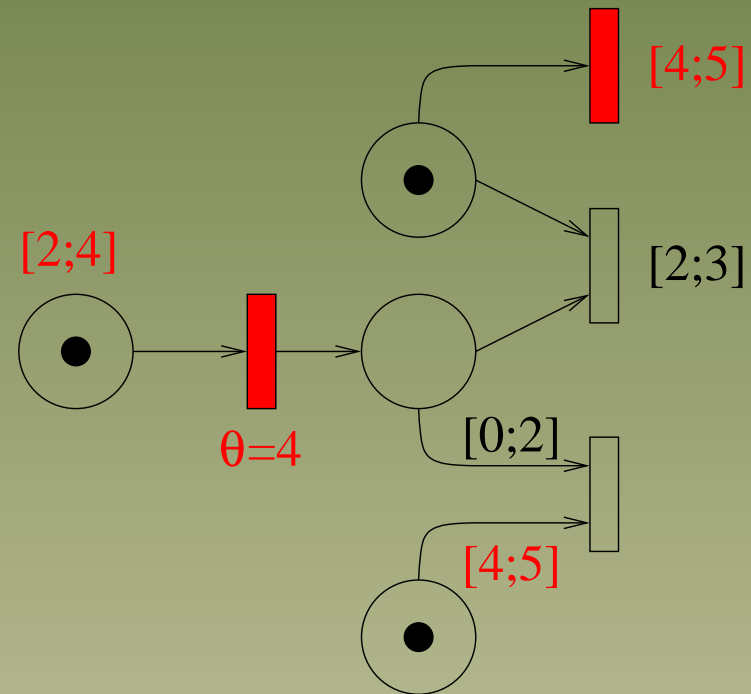
- transitions (t-RdP)
[MERLIN 74]
- places (p-RdP)
[KHANSA 97]
- arcs (a-RdP)
[WALTER 83]



(a.1) Réseaux de PETRI à intervalles

Réseaux de PETRI avec intervalles sur les :

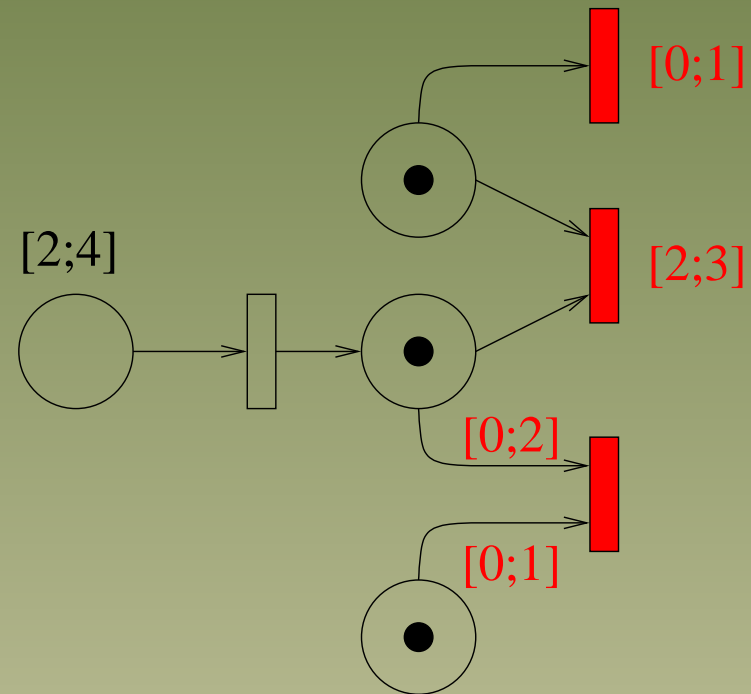
- transitions (t-RdP)
[MERLIN 74]
- places (p-RdP)
[KHANSA 97]
- arcs (a-RdP)
[WALTER 83]



(a.1) Réseaux de PETRI à intervalles

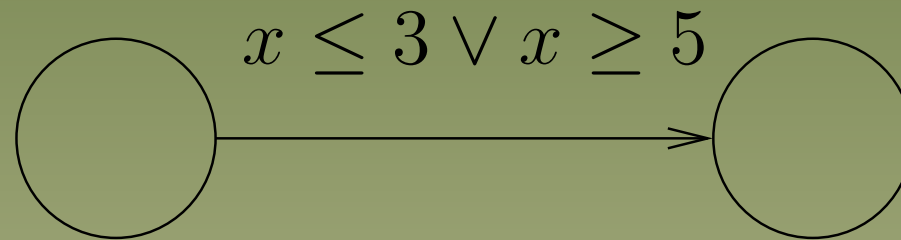
Réseaux de PETRI avec intervalles sur les :

- transitions (t-RdP)
[MERLIN 74]
- places (p-RdP)
[KHANSA 97]
- arcs (a-RdP)
[WALTER 83]



(a.2) Systèmes à gardes algébriques

Les conditions ne sont plus des intervalles, mais des expressions sur des *horloges*.



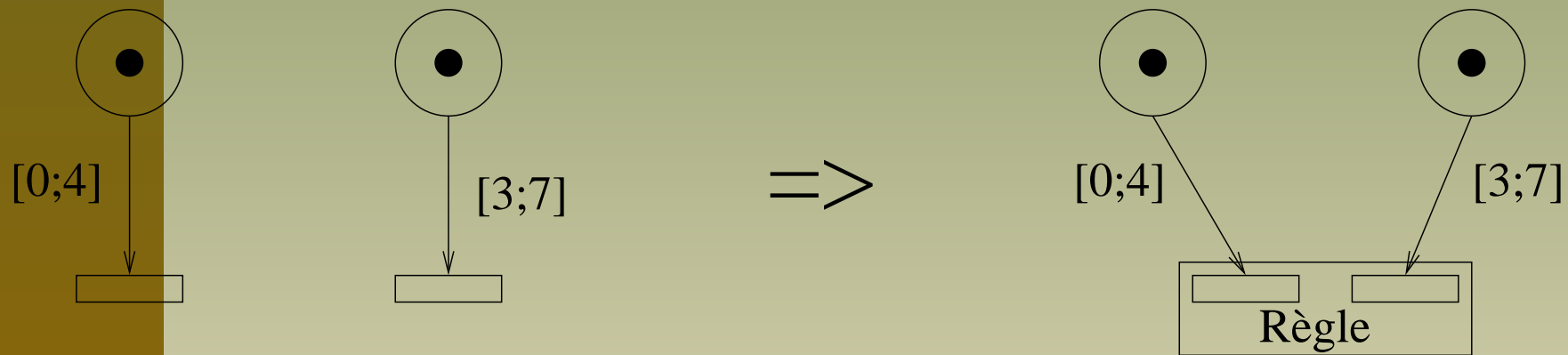
- système plus expressif ($[2; 3] \equiv 2 \leq x \wedge x \leq 3$)
- gestion explicite des horloges

(a.3) Études actuelles

Objectifs : permettre de spécifier un système comme synchronisation *temporelle* de plusieurs éléments

Moyens :

- règles de synchronisation (dernier arrivé, maître...) , *RdP* à flux temporels [DIAZ 94]

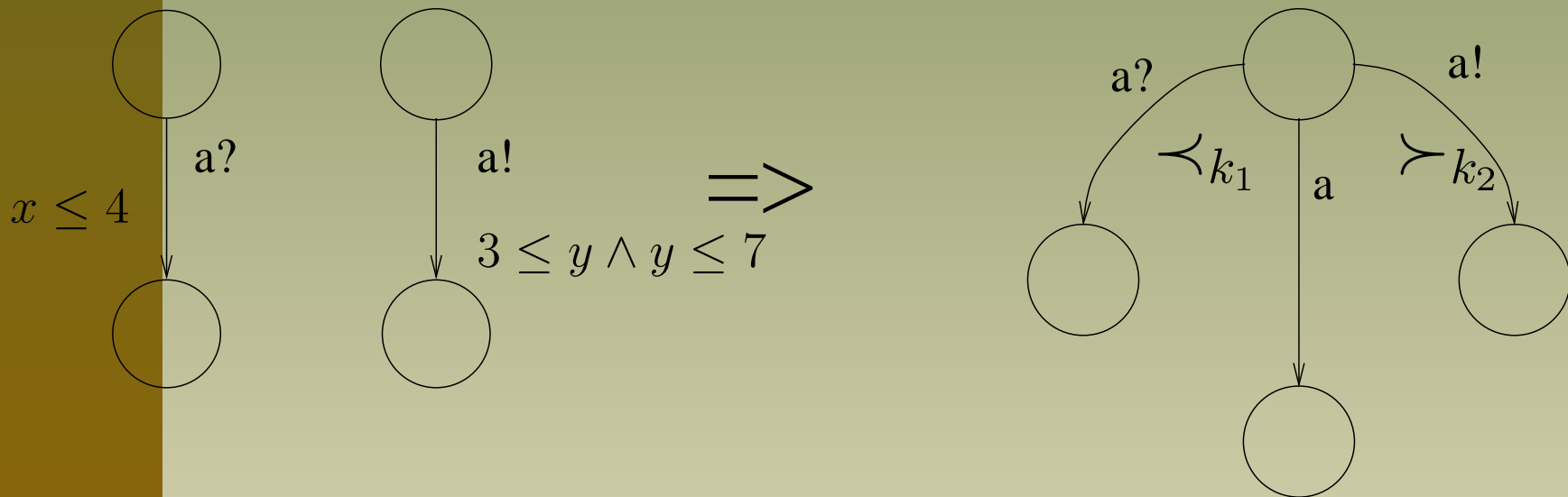


(a.3) Études actuelles

Objectifs : permettre de spécifier un système comme synchronisation *temporelle* de plusieurs éléments

Moyens :

- règles de synchronisation, *RdP à flux temporels* [DIAZ 94]
- priorités temporelles entre actions, *automates à urgence* [SIFAKIS 98]

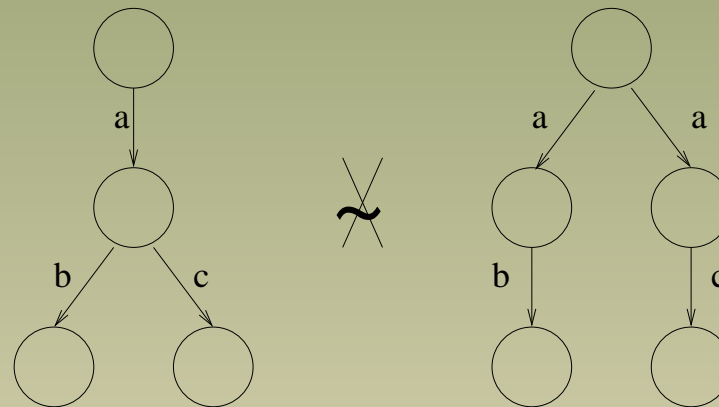


1. Modéliser les applications multimédias
2. **La question du temps...**
 - (a) État de l'art
 - (b) **Comparaisons des pouvoirs d'expressions**
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(b.1) Traductions et comparaisons

Critère de comparaison : bisimulation

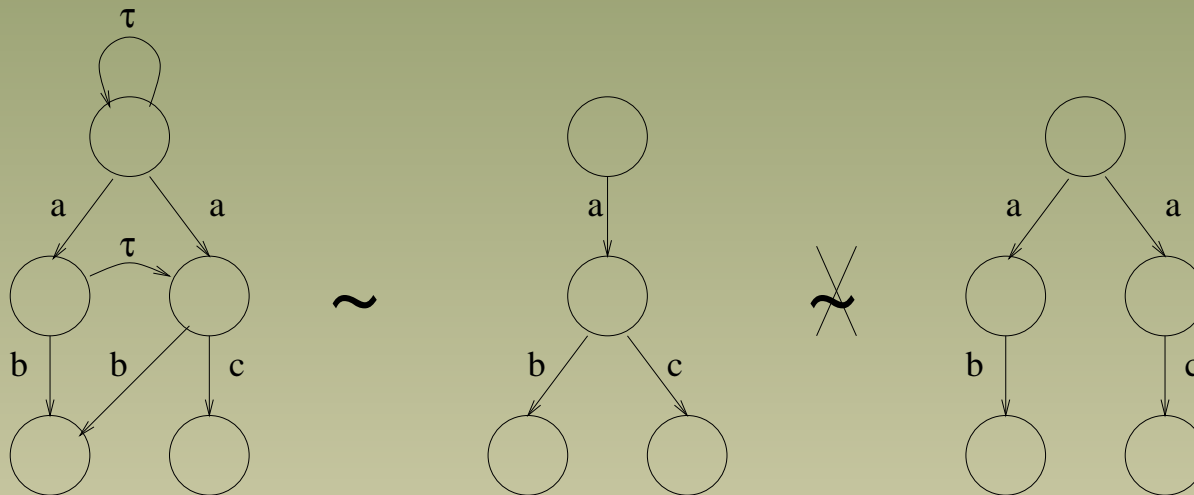
- bisimulation : mêmes comportements



(b.1) Traductions et comparaisons

Critère de comparaison : bisimulation faible

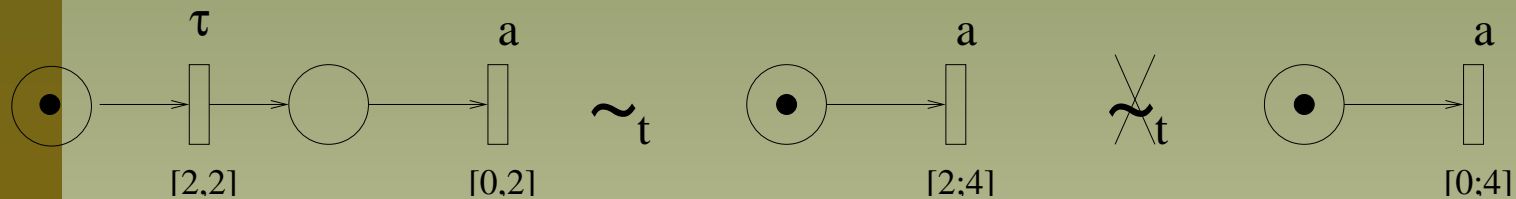
- bisimulation : mêmes comportements
- faible : mêmes comportements *visibles*



(b.1) Traductions et comparaisons

Critère de comparaison : bisimulation *temporelle* faible

- bisimulation : mêmes comportements
- faible : mêmes comportements *visibles*
- temporelle : mêmes comportements et mêmes durées



(b.1) Traductions et comparaisons

Critère de comparaison : bisimulation *temporelle* faible

- bisimulation : mêmes comportements
- faible : mêmes comportements *visibles*
- temporelle : mêmes comportements et mêmes durées

$$(p \sim_t q) \iff$$

$$(\forall a \in V \cup \{\tau\}, \forall \theta \in \mathbb{R}^+ : (p \xrightarrow{a,\theta} p') \Rightarrow (\exists q')(q \xrightarrow{a,\theta} q' \wedge p' \sim_t q'))$$

$$\wedge (\forall a \in V \cup \{\tau\}, \forall \theta \in \mathbb{R}^+ : (q \xrightarrow{a,\theta} q') \Rightarrow (\exists p')(p \xrightarrow{a,\theta} p' \wedge p' \sim_t q'))$$

(b.2) Entre les RdP à intervalles

Hiérarchie :

a-RdP

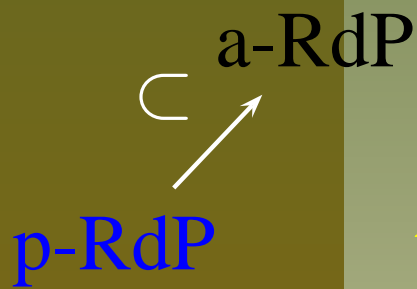
- p-RdP : intervalle sur places
- t-RdP : intervalle sur transitions
- a-RdP : intervalle sur arcs

p-RdP

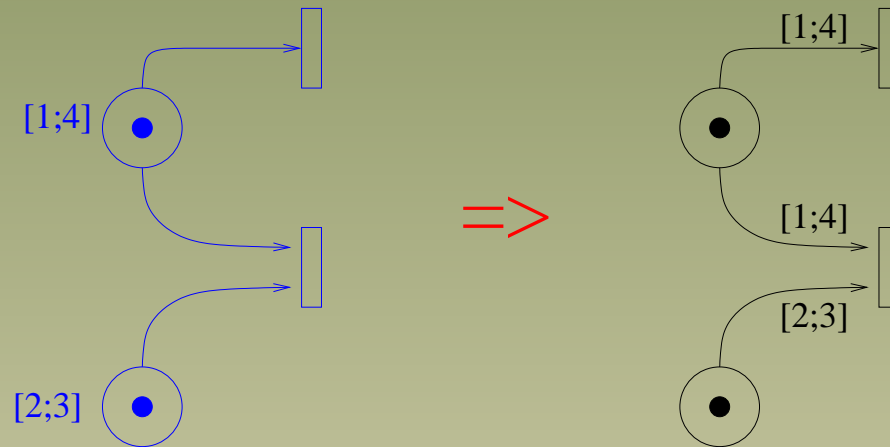
t-RdP

(b.2) Entre les RdP à intervalles

Hiérarchie :

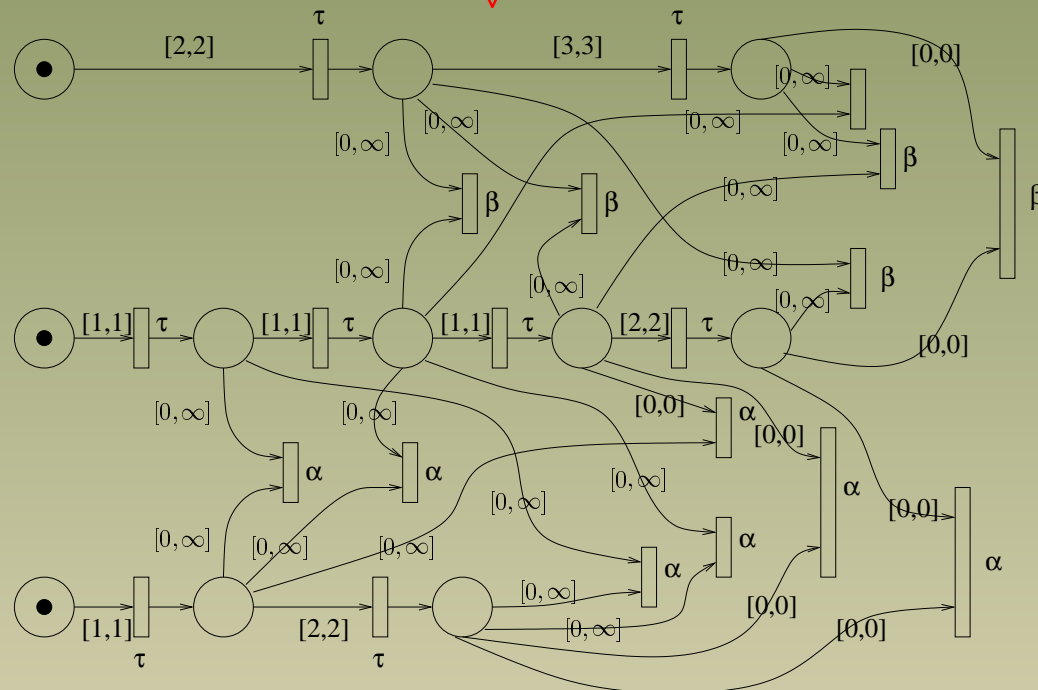
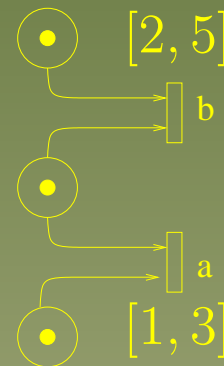
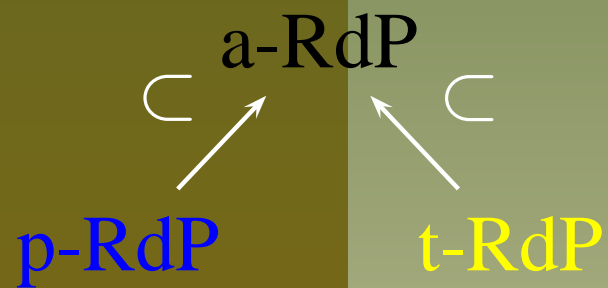


t-RdP



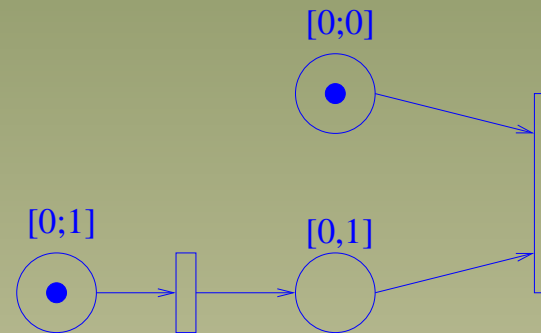
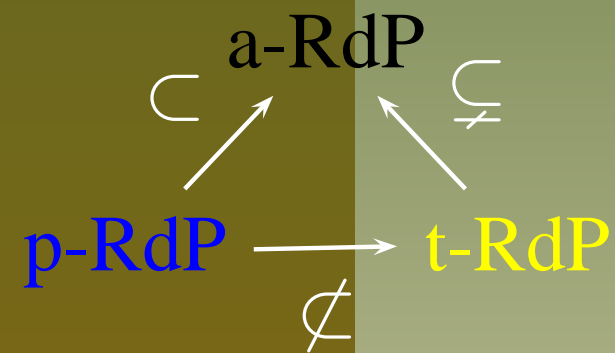
(b.2) Entre les RdP à intervalles

Hiérarchie :



(b.2) Entre les RdP à intervalles

Hiérarchie :



(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

- facile : car gardes sont très riches

(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

- facile : car gardes sont très riches
- uniquement dans le cas borné

(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

- facile : car gardes sont très riches
- uniquement dans le cas borné

Systèmes synchronisés sont seulement des abréviations :

(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

- facile : car gardes sont très riches
- uniquement dans le cas borné

Systèmes synchronisés sont seulement des abréviations :

- RdP à flux temporels \equiv RdP intervalles

(b.3) Systèmes algébriques : sur-modèle

Traduction de RdP à intervalles en systèmes à gardes algébriques (automates temporisés)

- facile : car gardes sont très riches
- uniquement dans le cas borné

Systèmes synchronisés sont seulement des abréviations :

- RdP à flux temporels \equiv RdP intervalles
- Automates à urgence \cong automates temporisés

1. Modéliser les applications multimédias
2. **La question du temps...**
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) **Comparaisons d'utilisations**
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :

- l'alarme est armée à une date t_0 avec une durée θ

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :

- l'alarme est armée à une date t_0 avec une durée θ
- si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :

- l'alarme est armée à une date t_0 avec une durée θ
- si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée
- sinon, à $t_0 + \theta$, l'alarme se déclenche

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :

- l'alarme est armée à une date t_0 avec une durée θ
- si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée
- sinon, à $t_0 + \theta$, l'alarme se déclenche

- Utilité : actions correctives

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :
 - l'alarme est armée à une date t_0 avec une durée θ
 - si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée
 - sinon, à $t_0 + \theta$, l'alarme se déclenche
- Utilité : actions correctives
- Nécessite dans le modèle

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :
 - l'alarme est armée à une date t_0 avec une durée θ
 - si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée
 - sinon, à $t_0 + \theta$, l'alarme se déclenche
- Utilité : actions correctives
- Nécessite dans le modèle
 - notion d'urgence

(c.1) Modéliser tous les comportements

Paradigme très important : l'alarme (ou chien de garde)

- Principe :
 - l'alarme est armée à une date t_0 avec une durée θ
 - si l'action à surveiller intervient avant la date $t_0 + \theta$, l'alarme est désarmée
 - sinon, à $t_0 + \theta$, l'alarme se déclenche
- Utilité : actions correctives
- Nécessite dans le modèle
 - notion d'urgence
 - une urgence par action (et non par état)

(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous

(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec

(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

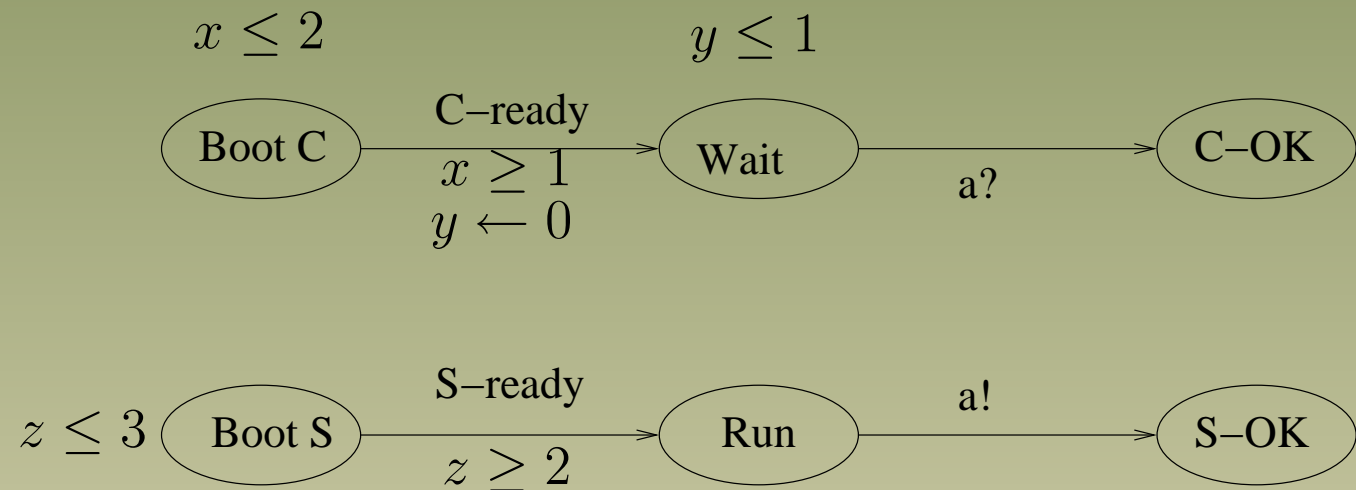
Exemple :

- Un serveur démarre en 2-3 s, puis offre un service a
- Un client démarre en 1-2 s, puis attend un service a , au plus 1s
- Rendez-vous impossible si le serveur démarre lentement et le client vite

(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

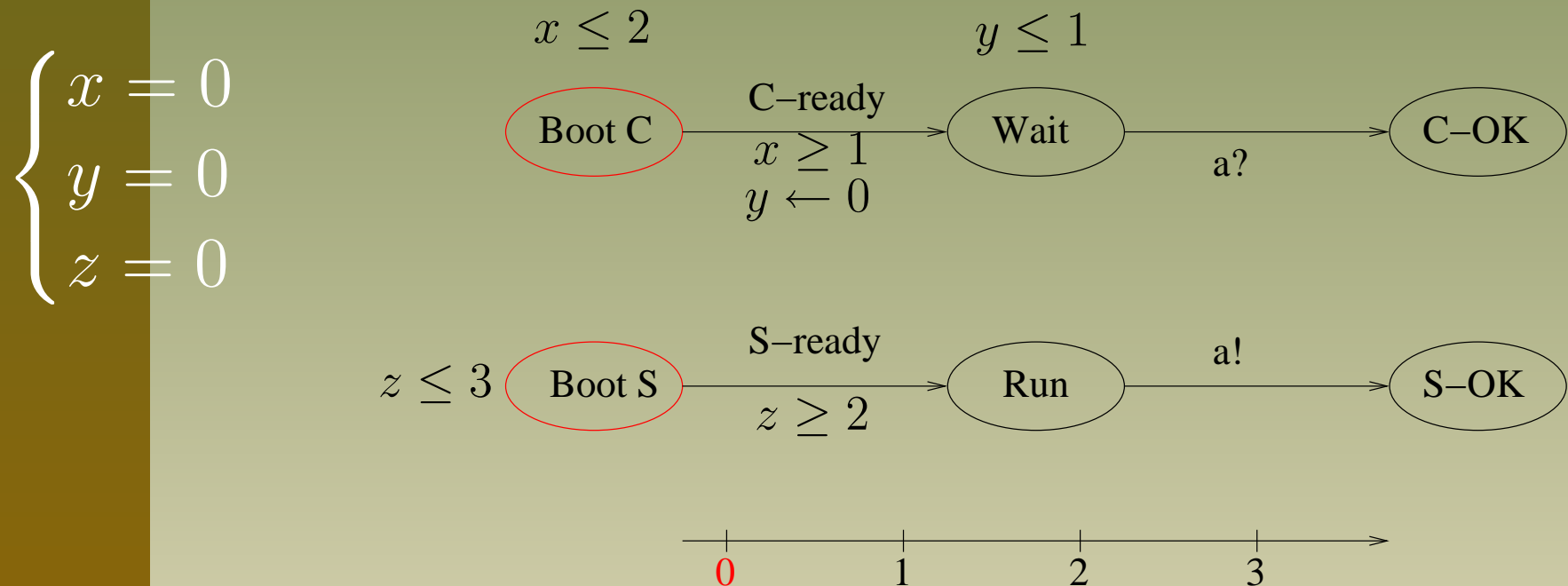
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

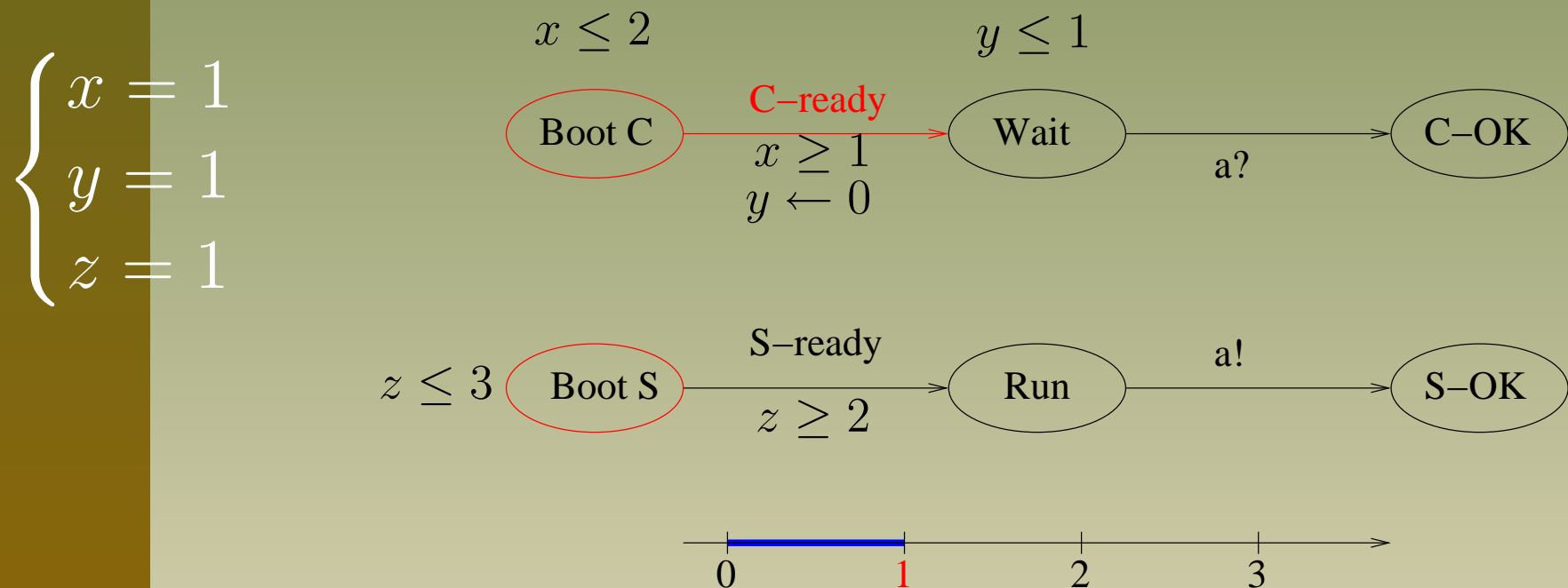
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

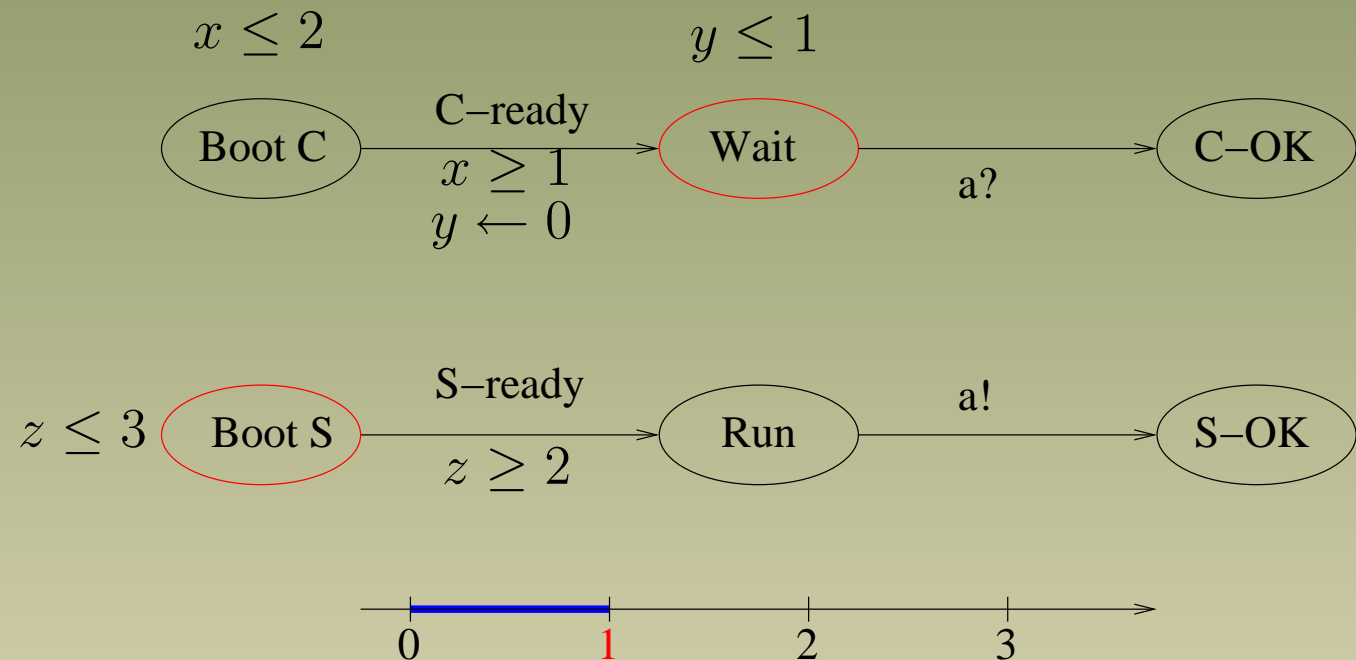


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

$$\begin{cases} x = 1 \\ y = 0 \\ z = 1 \end{cases}$$

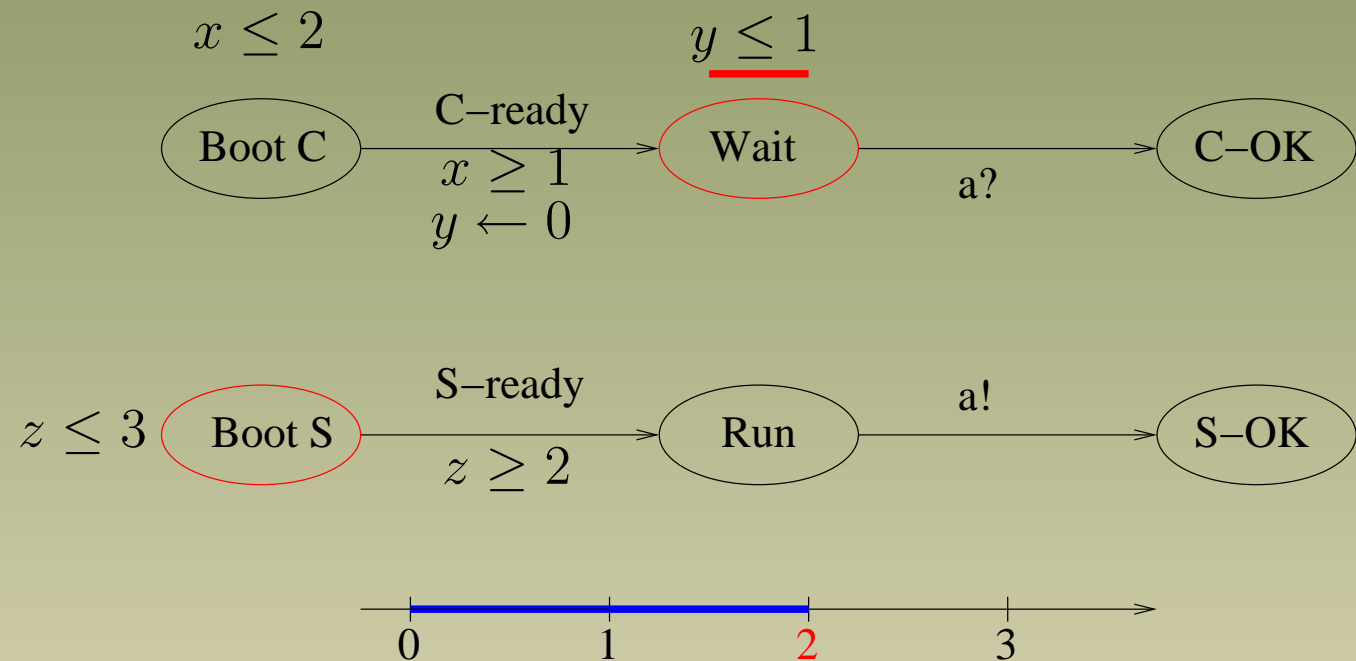


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

$$\begin{cases} x = 2 \\ y = 1 \\ z = 2 \end{cases}$$

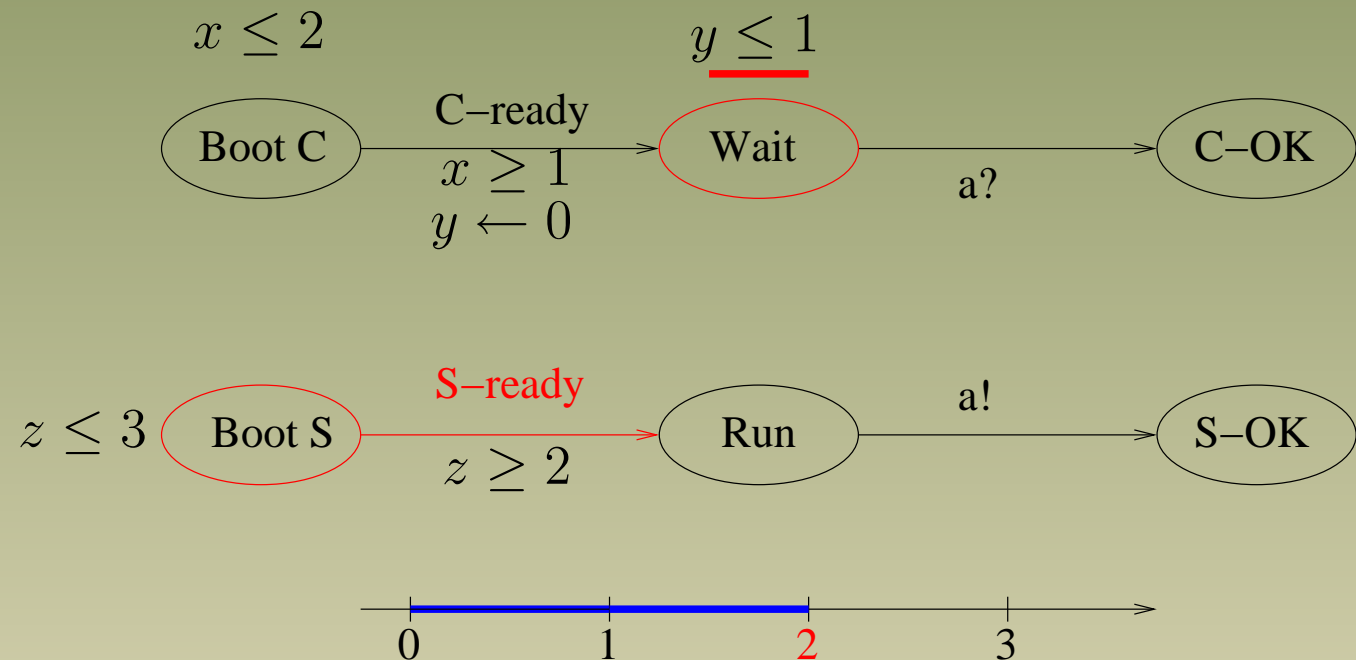


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

$$\begin{cases} x = 2 \\ y = 1 \\ z = 2 \end{cases}$$

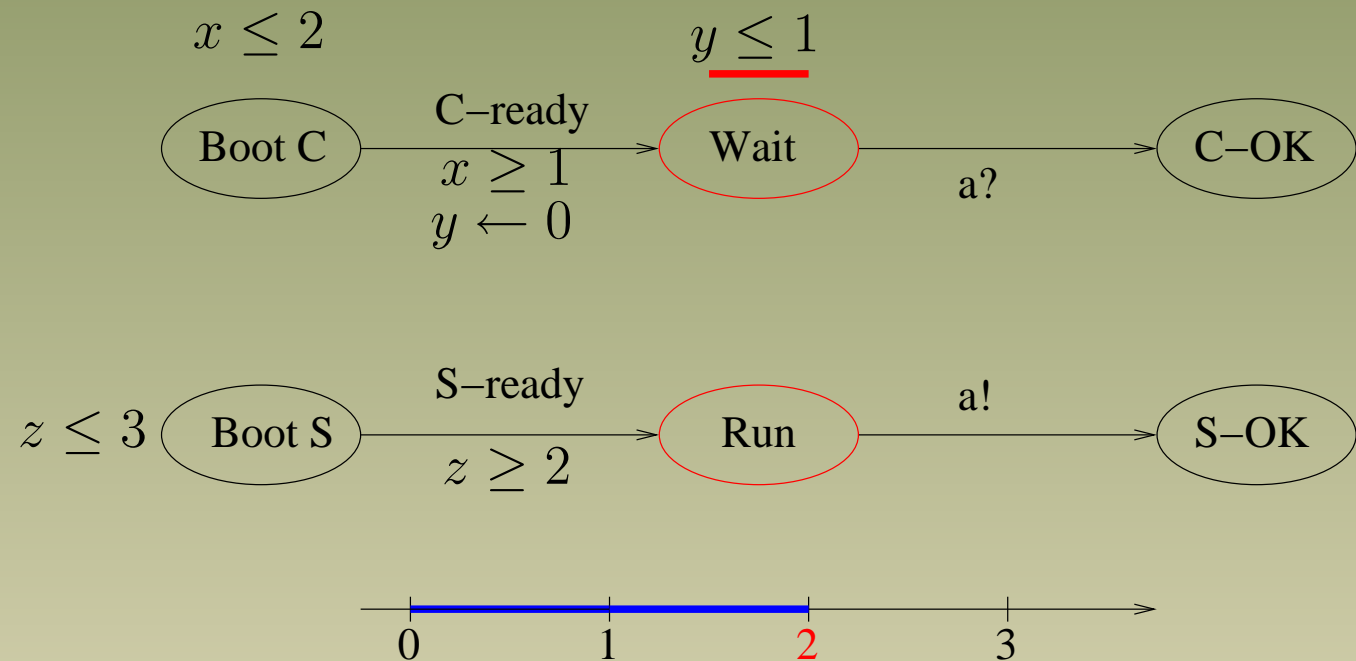


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

$$\begin{cases} x = 2 \\ y = 1 \\ z = 2 \end{cases}$$

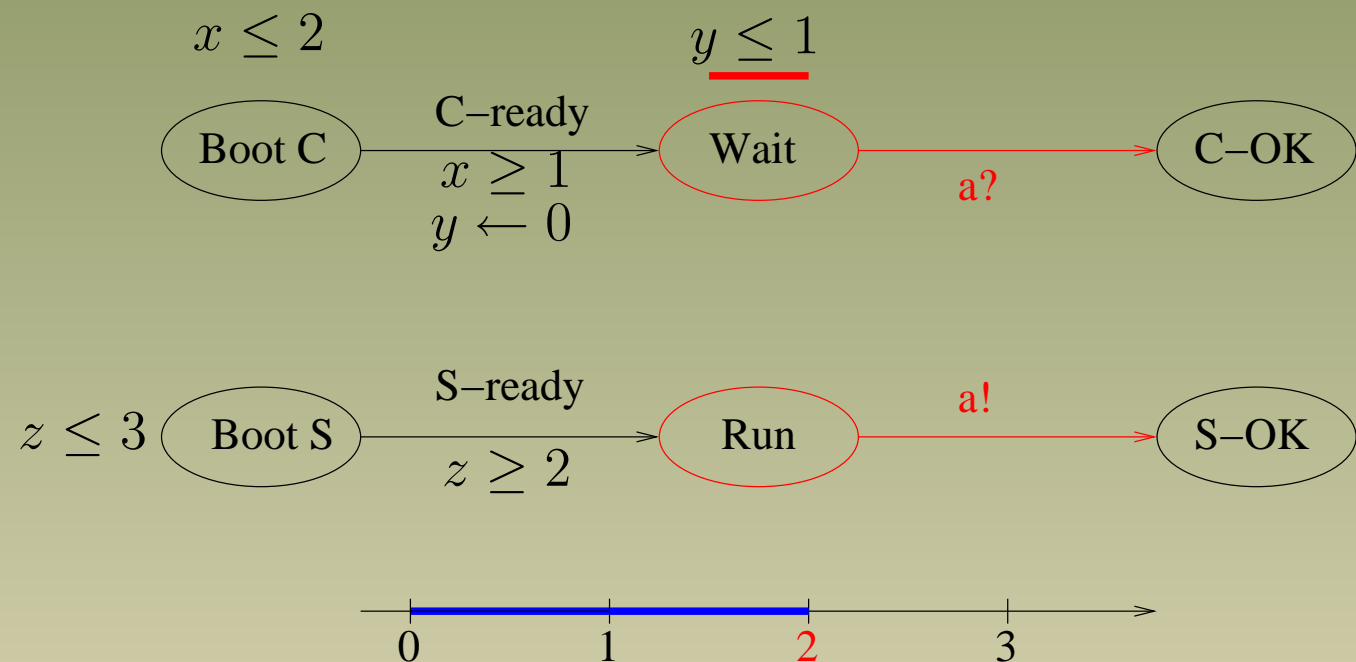


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

$$\begin{cases} x = 2 \\ y = 1 \\ z = 2 \end{cases}$$

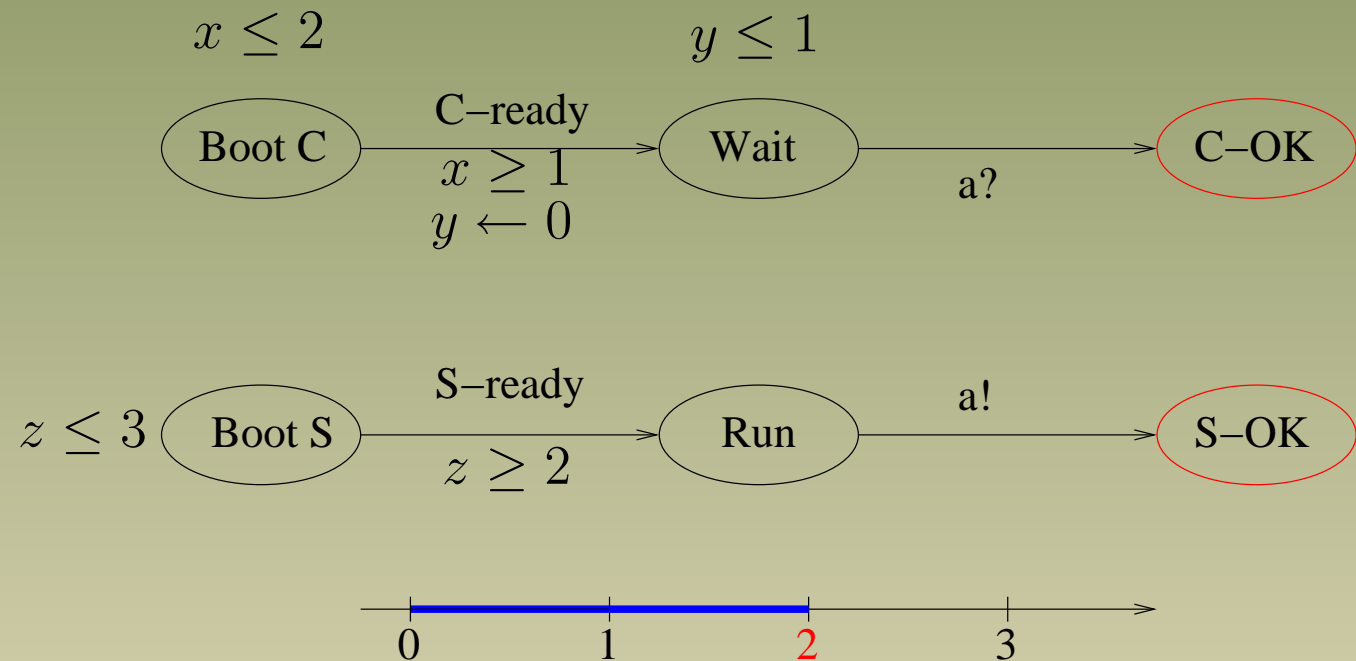


(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur

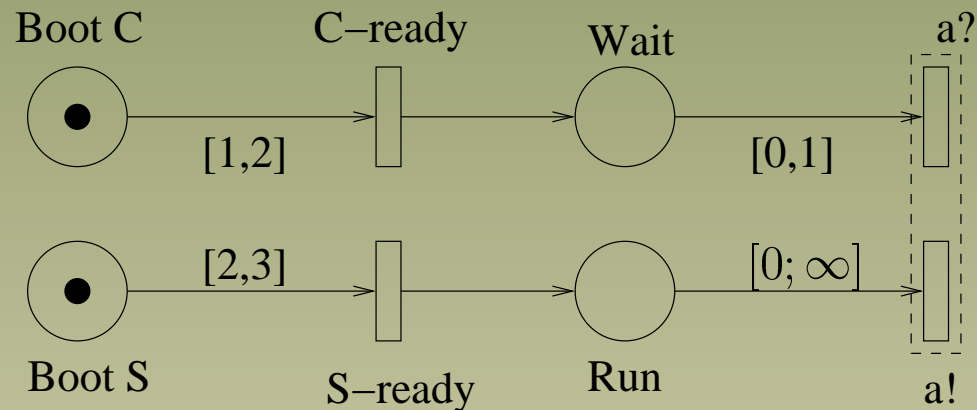
$$\begin{cases} x = 2 \\ y = 1 \\ z = 2 \end{cases}$$



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

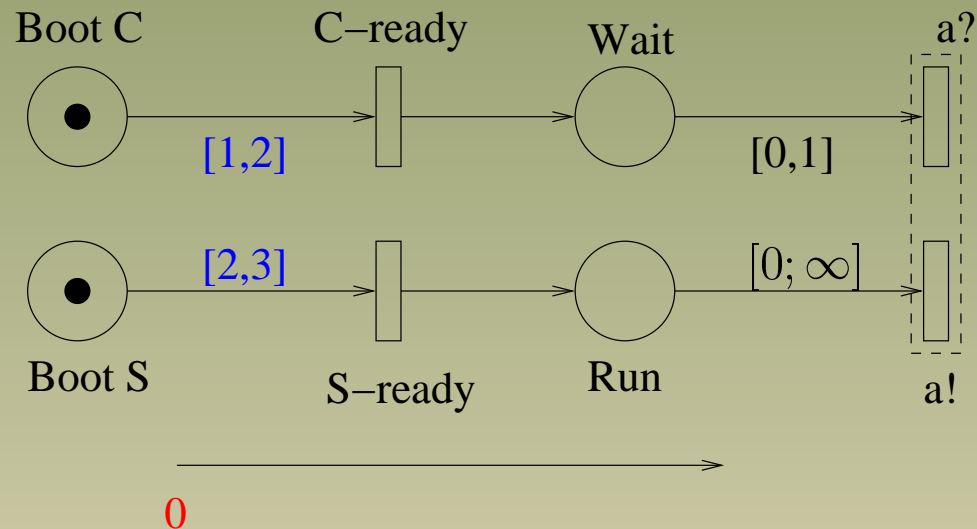
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

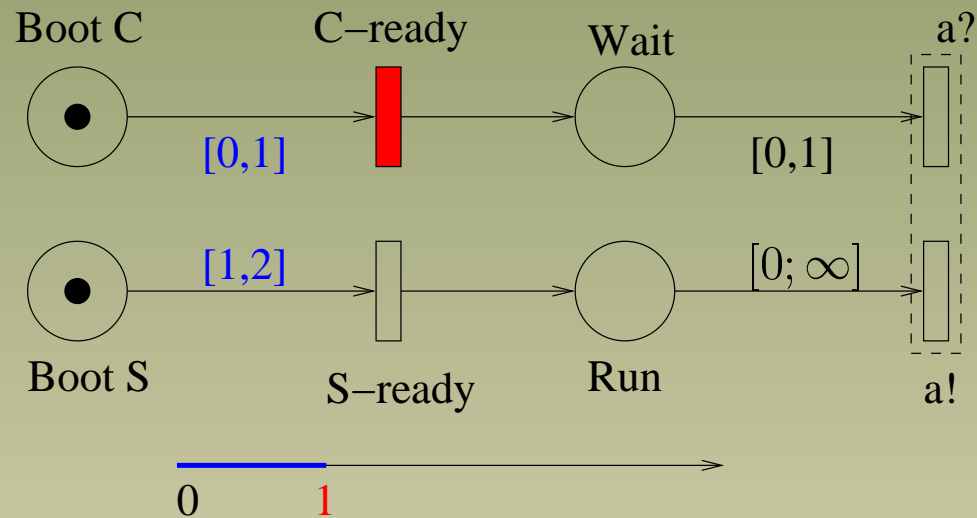
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

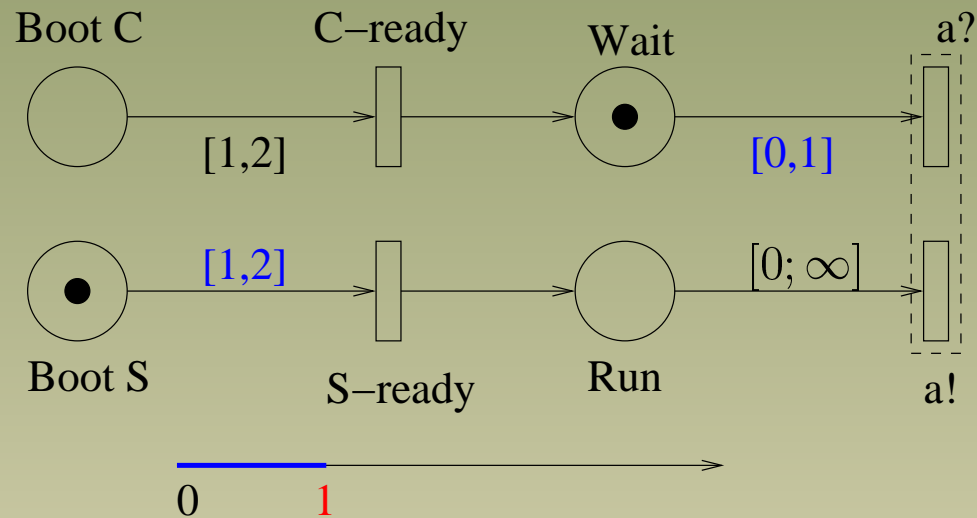
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

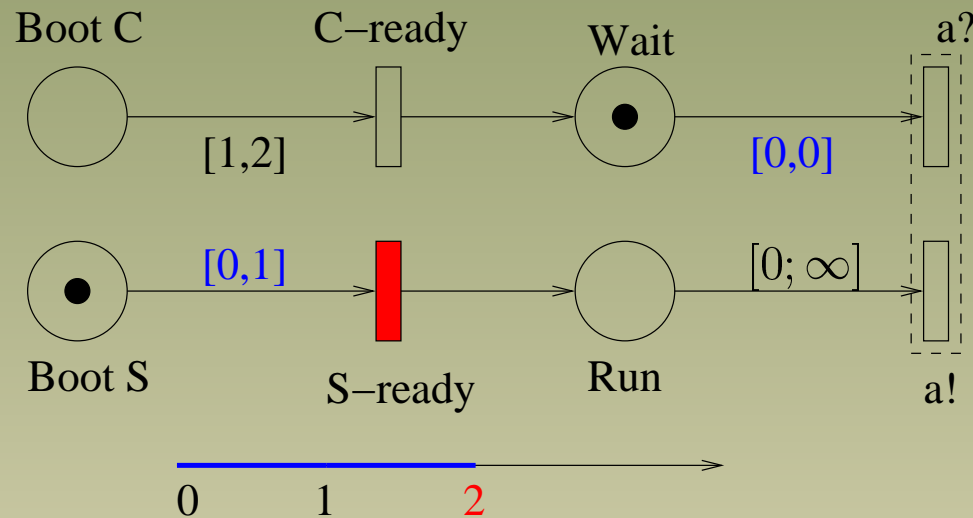
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

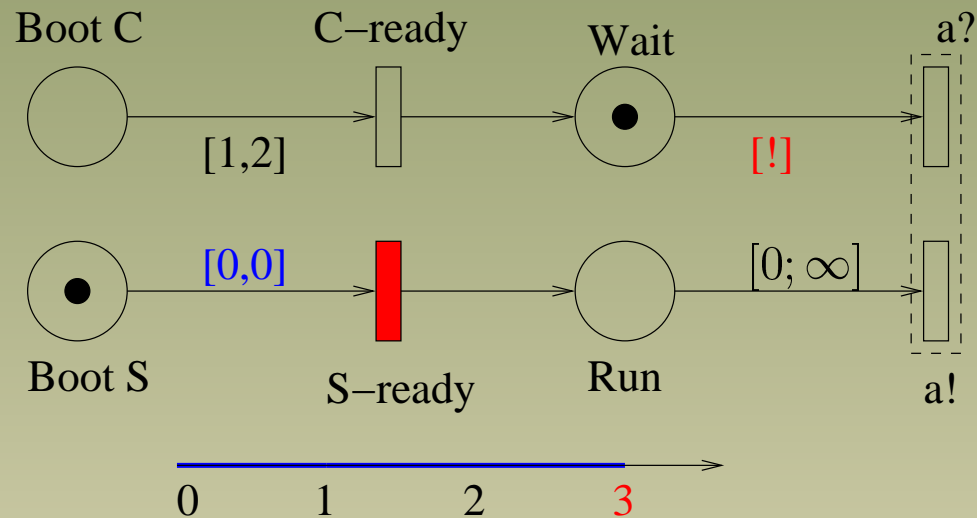
- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



(c.2) Modéliser les rendez-vous ratés

Synchronisation d'entités :

- système réel doit faire un rendez-vous
- modélisation doit pouvoir détecter l'échec
- modèle formel peut être trompeur



1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(a.1) Spécificité du multimédia

- Données typées par média
- Données structurées
- Contraintes temporelles
- Architecture distribuée

(a.1) Spécificité du multimédia

- Données typées par média
- Données structurées
- Contraintes temporelles
- Architecture distribuée

Contributions passées :

- RdP à flux temporel pour les données
- dessin de “boites” pour l’architecture
- aucun modèle pour les applications

(a.2) Objectifs du modèle

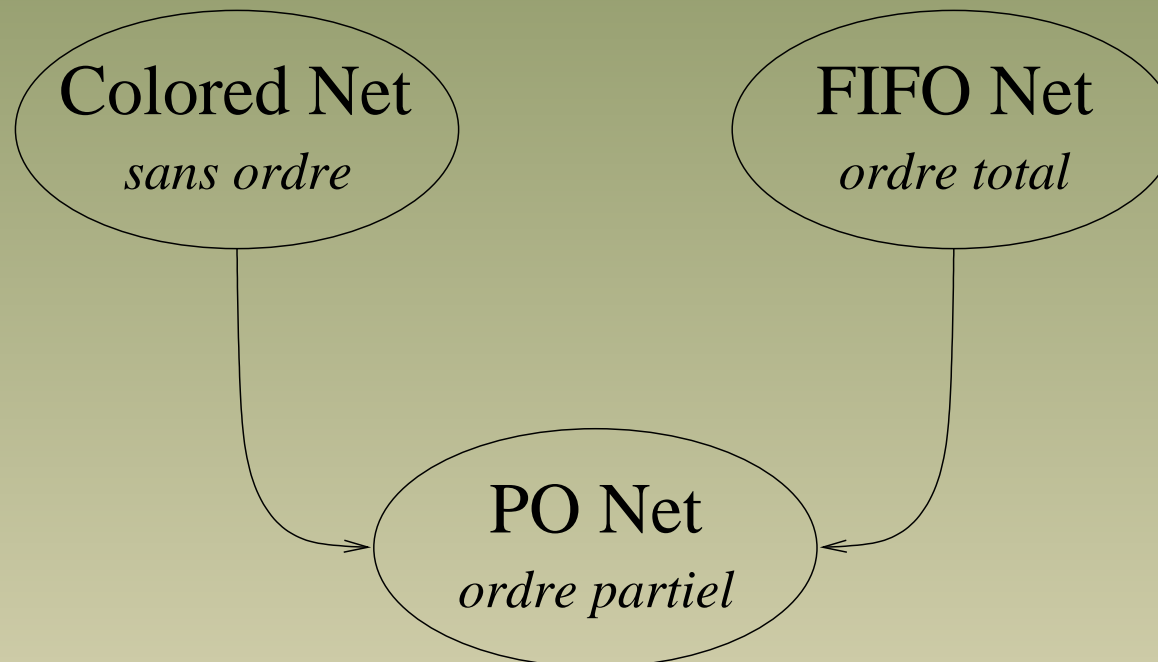
- Modéliser mais séparer
 - l'architecture
 - ⇒ aspect structurel
 - le traitement des données
 - ⇒ aspect dynamique
- Prendre en compte l'aspect temporel
 - ⇒ intégrer les résultats de l'étude

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(b.1) RdP à ordre partiel (Po-Net)

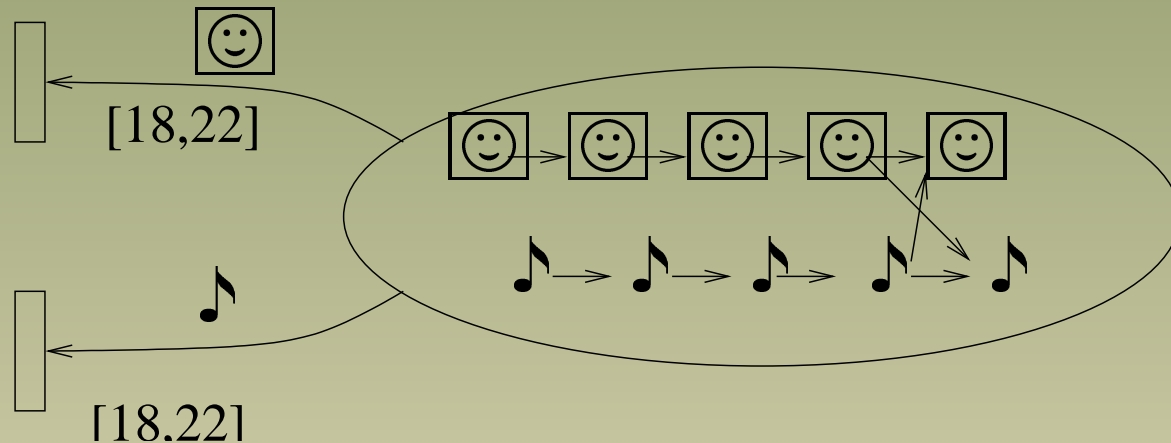
Premiers travaux :

- Tr-Net et Po-Net [FANCHON,BOYER 98], [FANCHON 99]
- RdP avec mots partiels comme marquage des places
- généralisent les RdP colorés et RdP à files



(b.2) TPONet : Adaptation au multimédia

- notion de type de média (♪, ☺)
- notion de flux (bande audio, vidéo) ordonnés + un flux non ordonné
- intervalles sur les arcs
- synchronisation par règles RdP à flux temporels



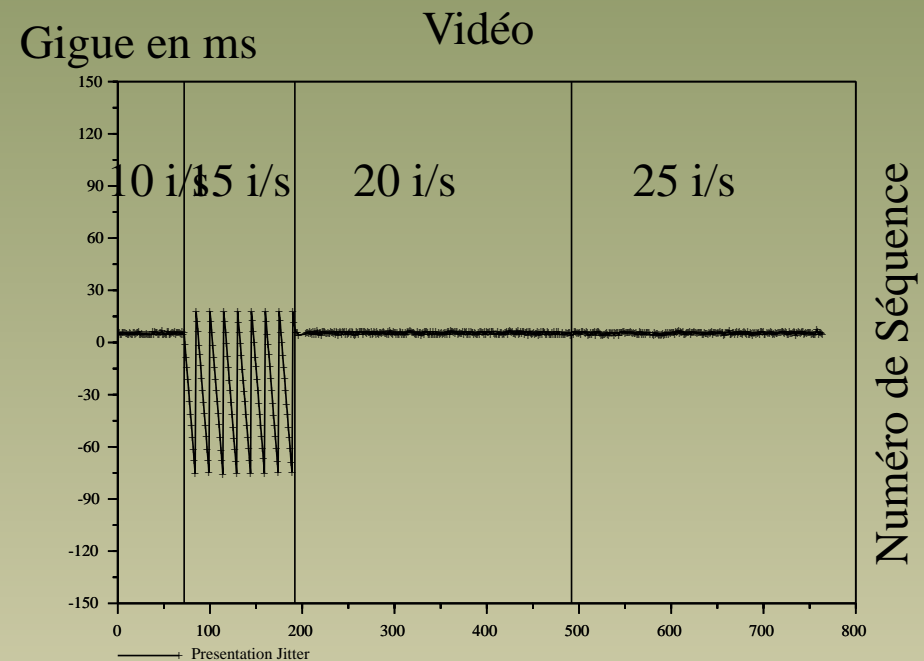
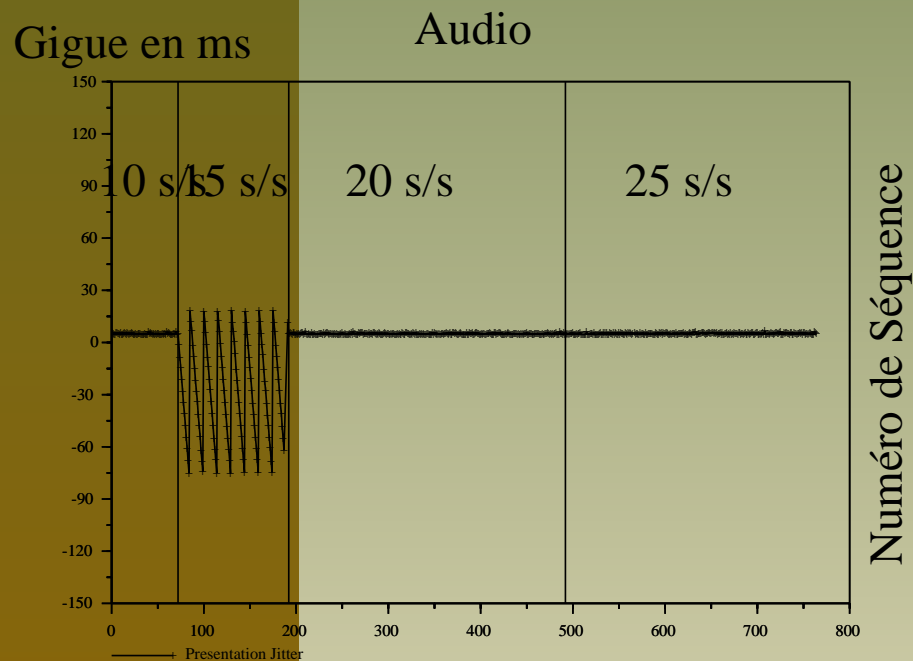
1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

(c.1) Système de visio-conférence

- PNSVS : PN Synchronised Videoconferencing System
 - Système émetteur / récepteur
 - Synchronisation des flux audios et vidéo
 - Renégociation dynamique de QoS
- Implémentation :
 - 9 threads, 21 zones d'exclusion mutuelle
 - synchronisation garantie à 10ms, moyenne à 3ms

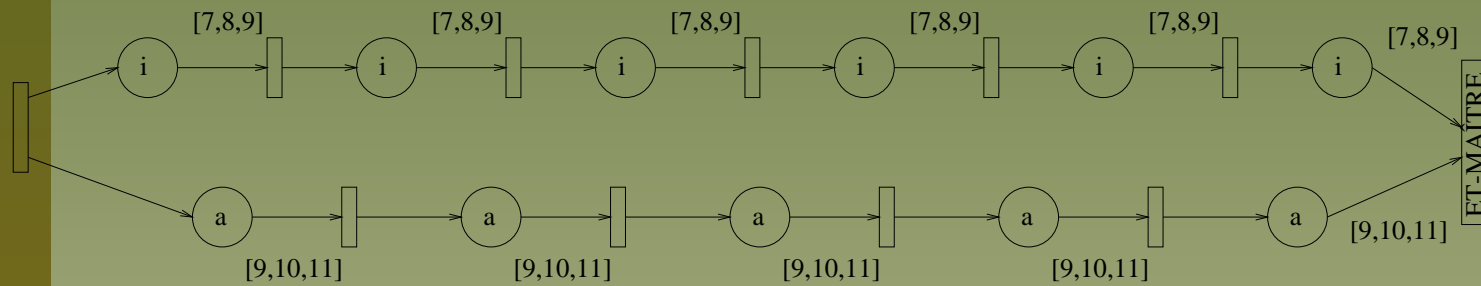
(c.1) Système de visio-conférence

- PNSVS : PN Synchronised Videoconferencing System
- Implémentation :
 - 9 threads, 21 zones d'exclusion mutuelle
 - synchronisation garantie à 10ms, moyenne à 3ms

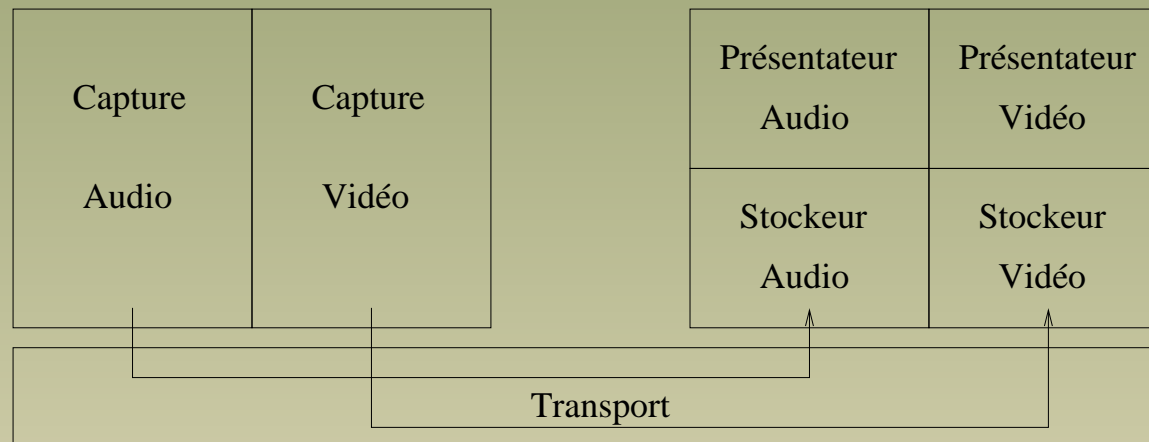


(c.2) Modélisations antérieures

■ Présentation (périodique) :

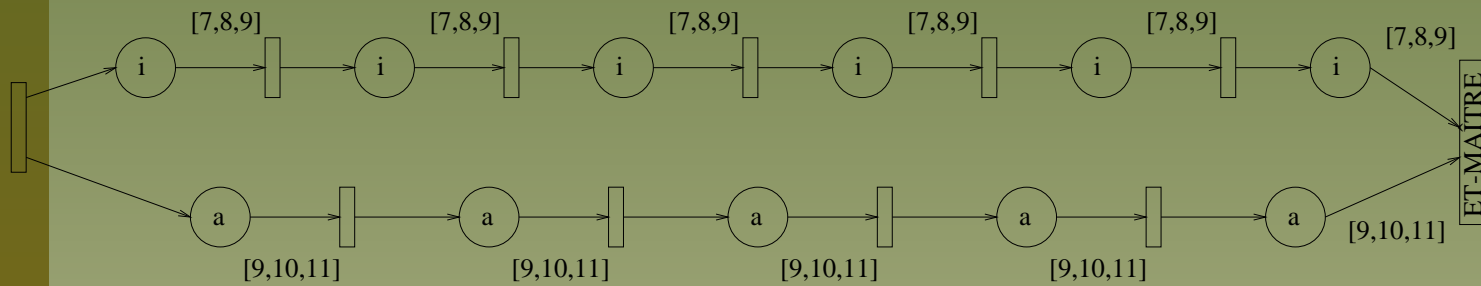


■ Architecture :

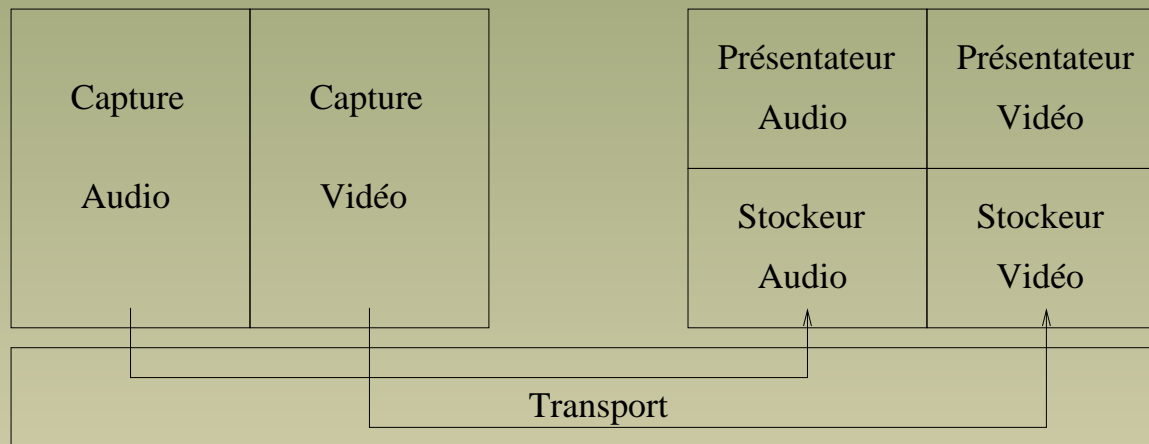


(c.2) Modélisations antérieures

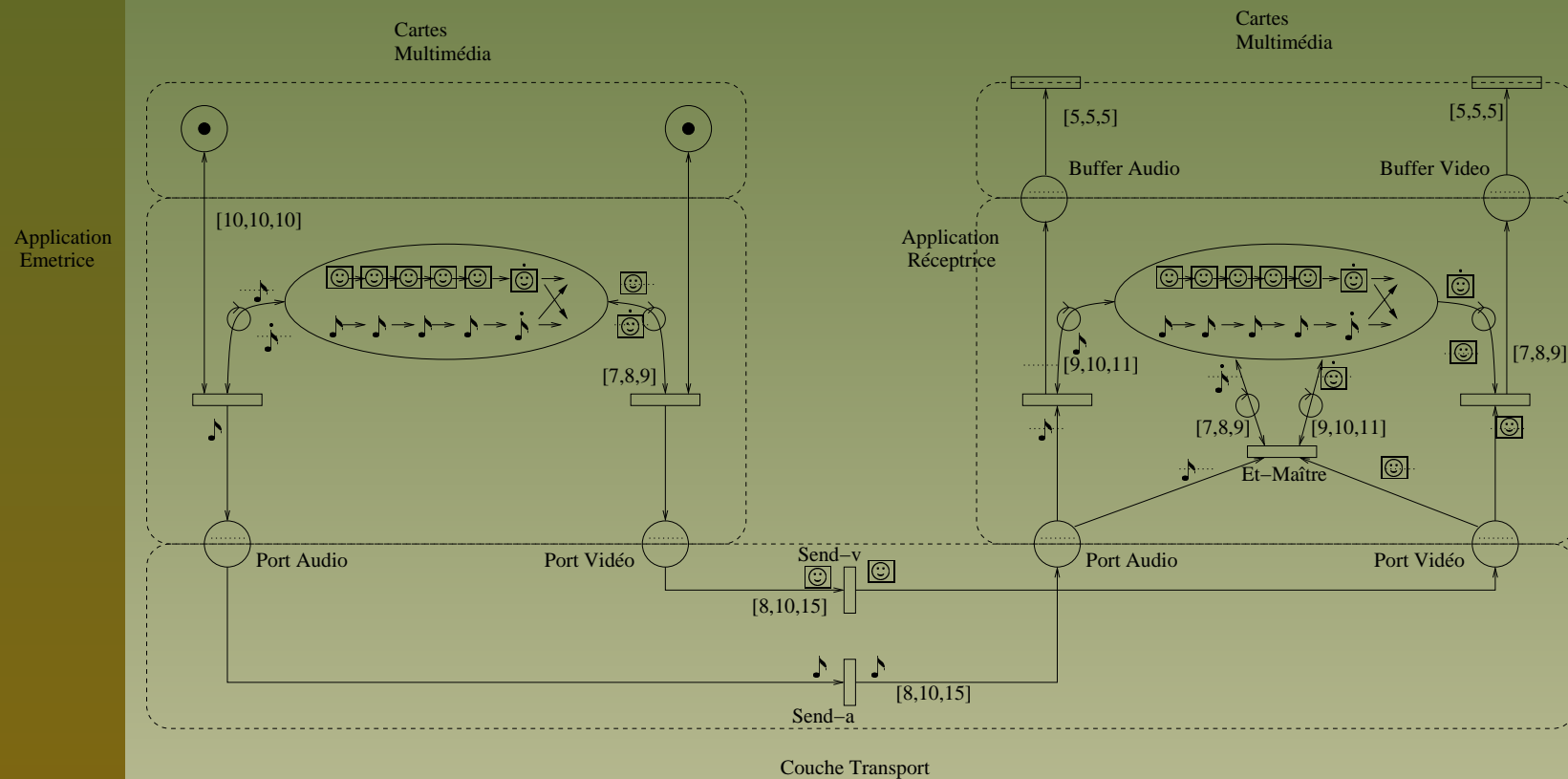
- Présentation (périodique) : marquage



- Architecture : structure



(c.3) Modélisation avec TPNNet



(c.4) Retour de l'expérience

Modèle adapté :

- modélisation application complète
- contenir la complexité
- sépare architecture et traitement données
- modélisation QdS dynamique

1. Modéliser les applications multimédias
2. La question du temps...
 - (a) État de l'art
 - (b) Comparaisons des pouvoirs d'expressions
 - (c) Comparaisons d'utilisations
3. ... et celles du multimédia
 - (a) Spécificité du multimédia
 - (b) Réseaux de PETRI à ordre partiel temporisé
 - (c) Exemple : application de visio-conférence
4. Conclusion et perspectives

Conclusion

Étude temporelle :

- comparaisons avec bases formelles
- liens entre modèles
- avantages et inconvénients

RdP temporisés à ordre partiel : modélisation intégrée

- architecture statique : places / transitions / arcs
- traitement des données : marquage
- contraintes temporelles : intervalles + règles

Perspectives

RdP temporisés à ordre partiel :

- traitement d'autres applications
- adaptation de méthodes de vérification
- outil

Conception de systèmes temporels :
intégration dans cadre spécification

- Projet RNTL COTRE (HOOD, UML)
- PostDoc Stanford (SDL)