Projet ProgConc [1]

Objectifs:

- Développer une application multithreading combinée à une approche RDP
- Analyser un paysage semi-complexe
- Versions évolutives de l'application pour améliorer la qualité du produit.
  Phase 1 obligatoire, les autres … bonus
  - Qualité de l'expressivité (impacte maintenance, évolution du système)
  - Sécurité
  - Robustesse
  - Performances

Choix du sujet: Carrefour routier avec gestion de deux flux

Phase 1:

- Cahier des charges
- Schématisation du paysage et des relations entre entités
- Listing des entités (comportements modélisables ou pas avec un RdP)
- Attention particulière pour le Timer
- Multithreading: version avec verrous de base (méthodes et/ou blocs `synchronized` combinés ou pas aux méthodes `wait()` et `notify()`)

Projet ProgConc [2]

- Cahier des charges modélisé avec RdP complet (deux modèles fournis)
  - Modèle avec Timer de base intégré au système de gestion
  - Modèle avec Timer évolué intégré ou pas au système de gestion
  - Trouver la faille conceptuelle du second modèle et améliorer!

- Implémentation du RDP:
  - Choix d'une architecture du RdP
  - Choix structure de données
  - Implémentation du Moteur − gestion des TICs en trois phases
  - Interfaçage RdP et son environnement

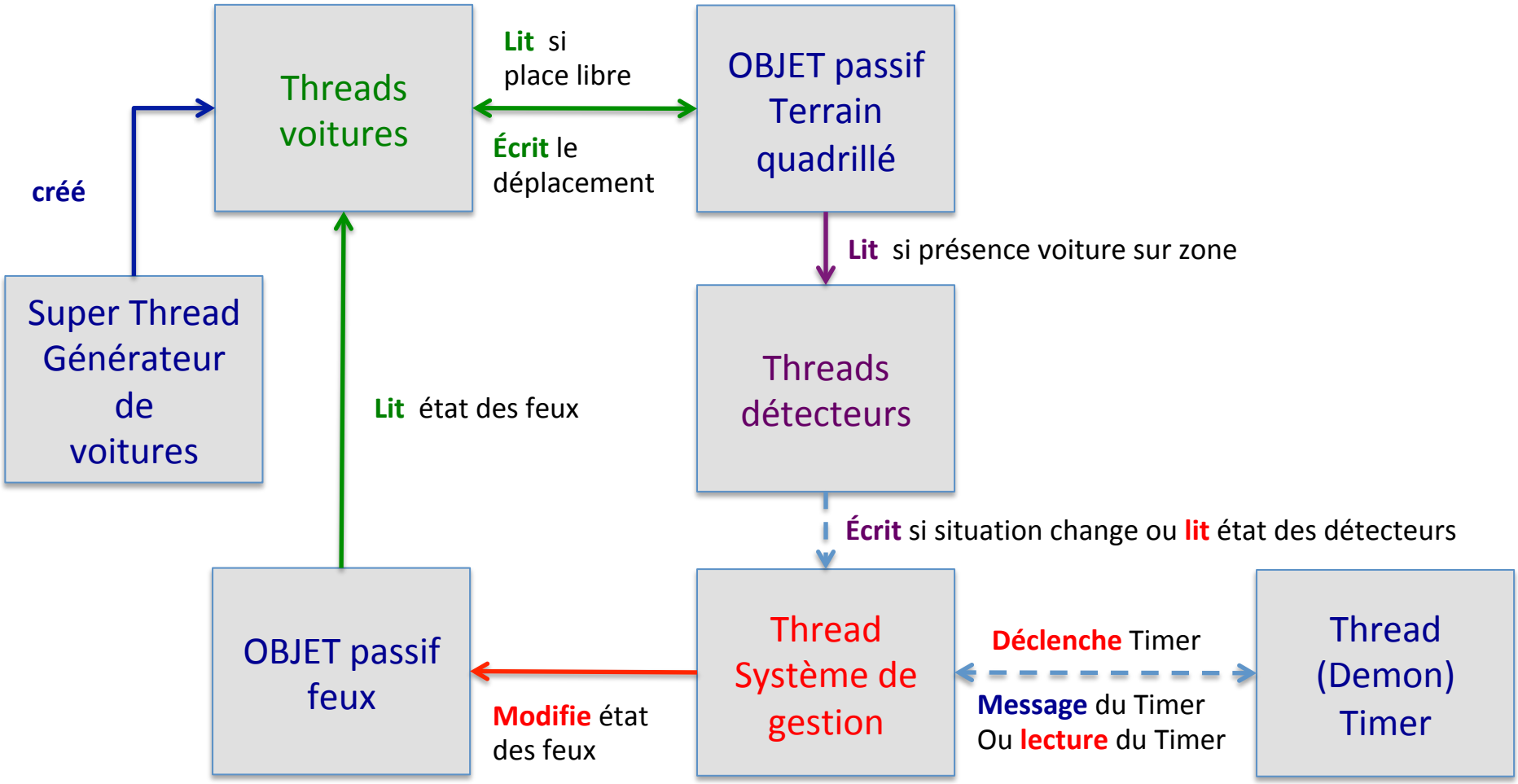
- Phase 2 et suivantes (optionnelles):
  - Justifier et exploiter les locks
  - Améliorations successives (justifier et implémenter)
  - Ou évolution du système (exemple: mode économique)

Paysage [1]

- Terrain quadrillé. Pour chaque flux
  - N_{C1} carrés sur route précédent la zone de détection
  - N_{C2} carrés sur zone de détection proximité du feu
  - N_{C3} carrés sur zone de détection présence dans le carrefour
- NB voitures générées par un générateur de flux (2 flux)
- 2 Générateurs de flux (avec créneaux horaires en option)
- 2 feux (avec deux états: Rouge/Vert)
- 4 détecteurs: 2 par flux
  - détection véhicules à proximité du feu,
  - véhicules dans carrefour
- Système de gestion du carrefour
- Timer intégré ou pas dans le système. A DISCUTER!

Paysage [2]

Le code des couleurs indique qui agit sur qui



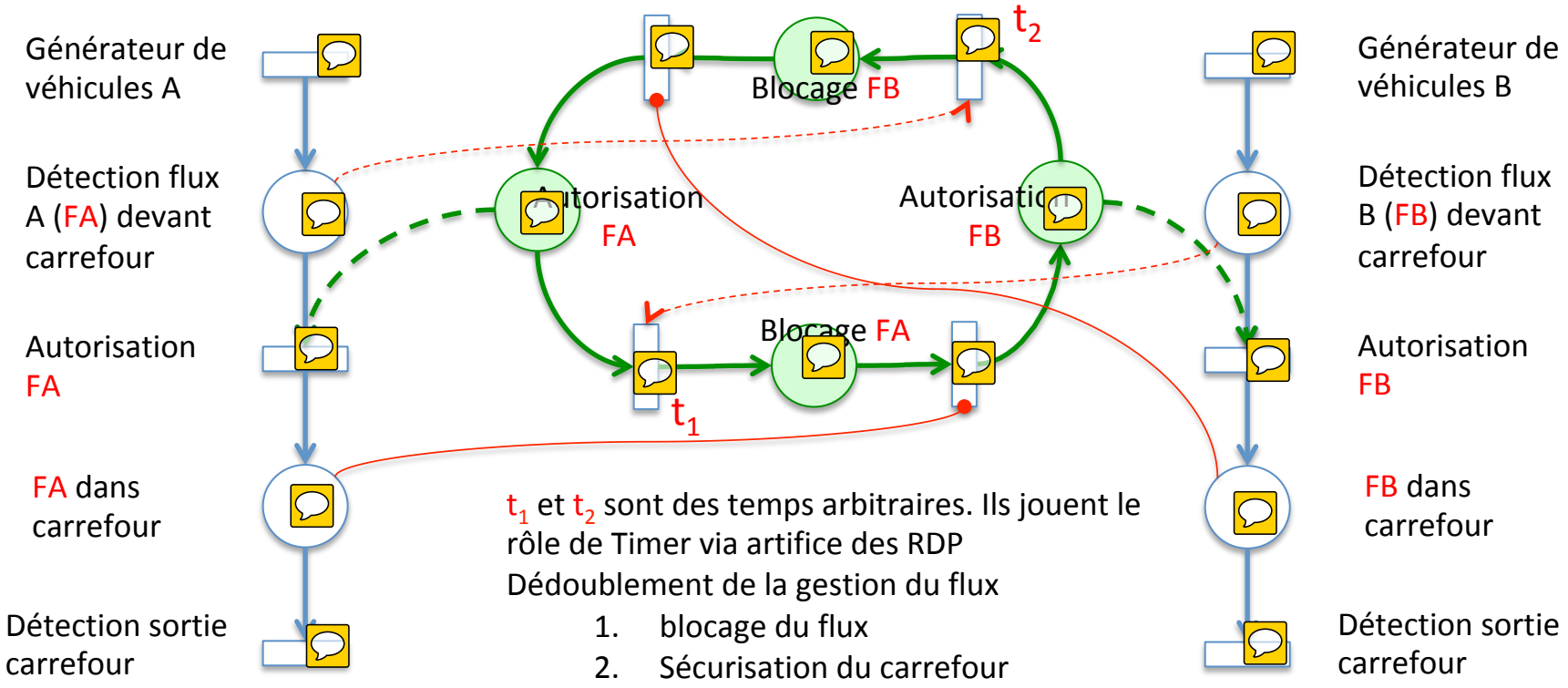
Explications sur le Timer

- Mode OFF : Aucune action n'est déclenchée
  - On reste dans ce mode si:
    - Un flux passe et l'autre inexistant
    - Les deux flux sont inexistants

- Mode ON : Déclenche un chronomètre
  - on passe de OFF à ON si
    - Flux A passe et Flux B est détecté
    - Flux B passe et Flux A est détecté

- Mode END : Bloque le flux A si occupant ou le flux B si occupant
  - On passe de ON à END si
    - Le temps imparti a été atteint par le chronomètre
    - Le temps imparti n'est pas atteint mais le flux A occupant est tari
    - Le temps imparti n'est pas atteint mais le flux B occupant est tari
  - On passe de END à OFF si:
    - Le flux A occupant a évacué le carrefour
    - Le flux B occupant a évacué le carrefour

Gestion du carrefour avec Timer intégré



t_1 et t_2 sont des temps arbitraires. Ils jouent le rôle de Timer via artifice des RDP

Dédoublement de la gestion du flux

1. blocage du flux
2. Sécurisation du carrefour

Avantages:

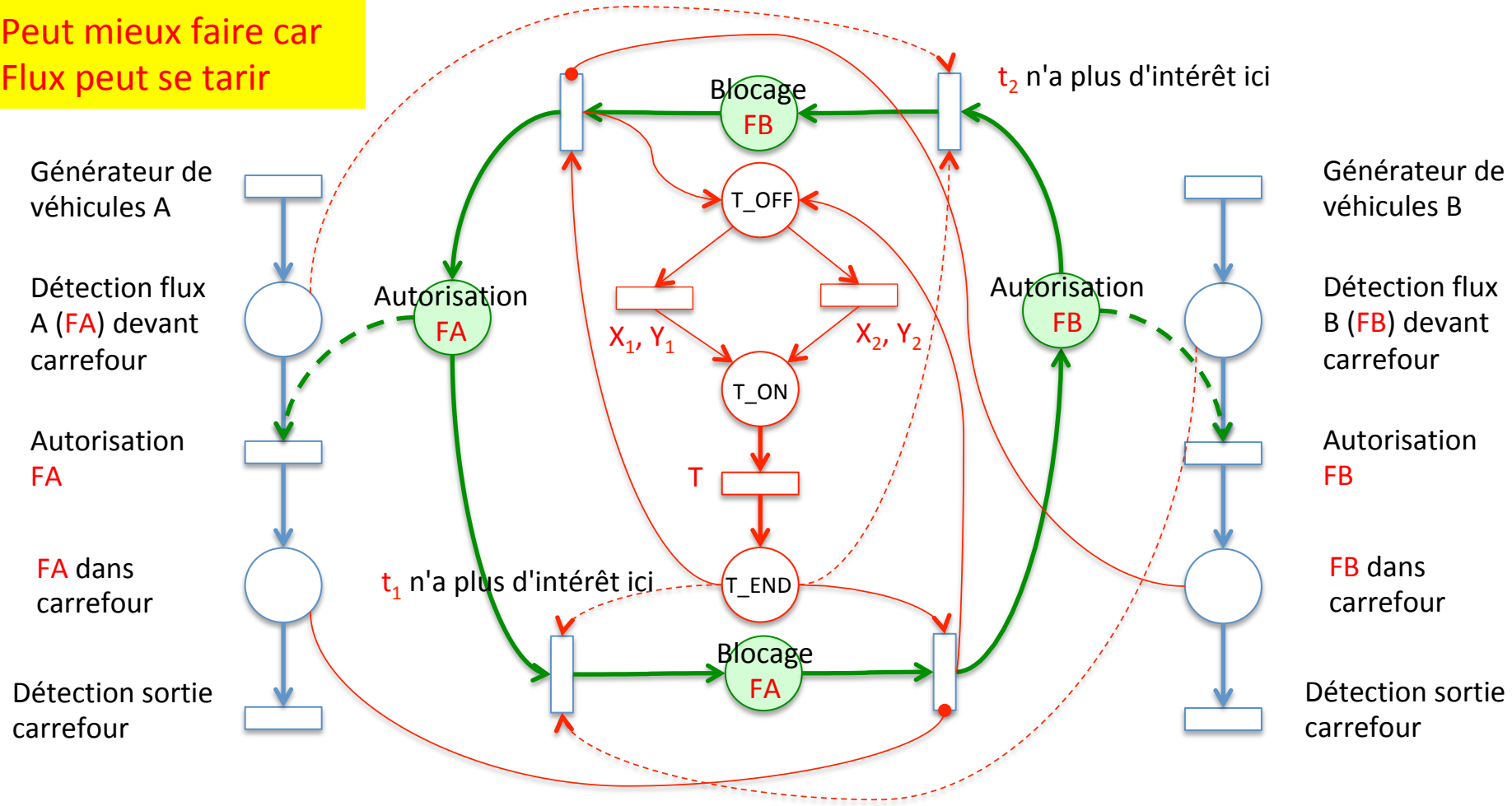
- Plus de basculement inutile (seulement lorsque le flux concurrent se manifeste)

Inconvénients:

- Inefficacité si le flux A (respectivement B) se tarît, alors on attend inutilement le temps imparti avant basculement (idem version 1)

Gestion du carrefour avec Timer évolué (V2)

Peut mieux faire car
Flux peut se tarir

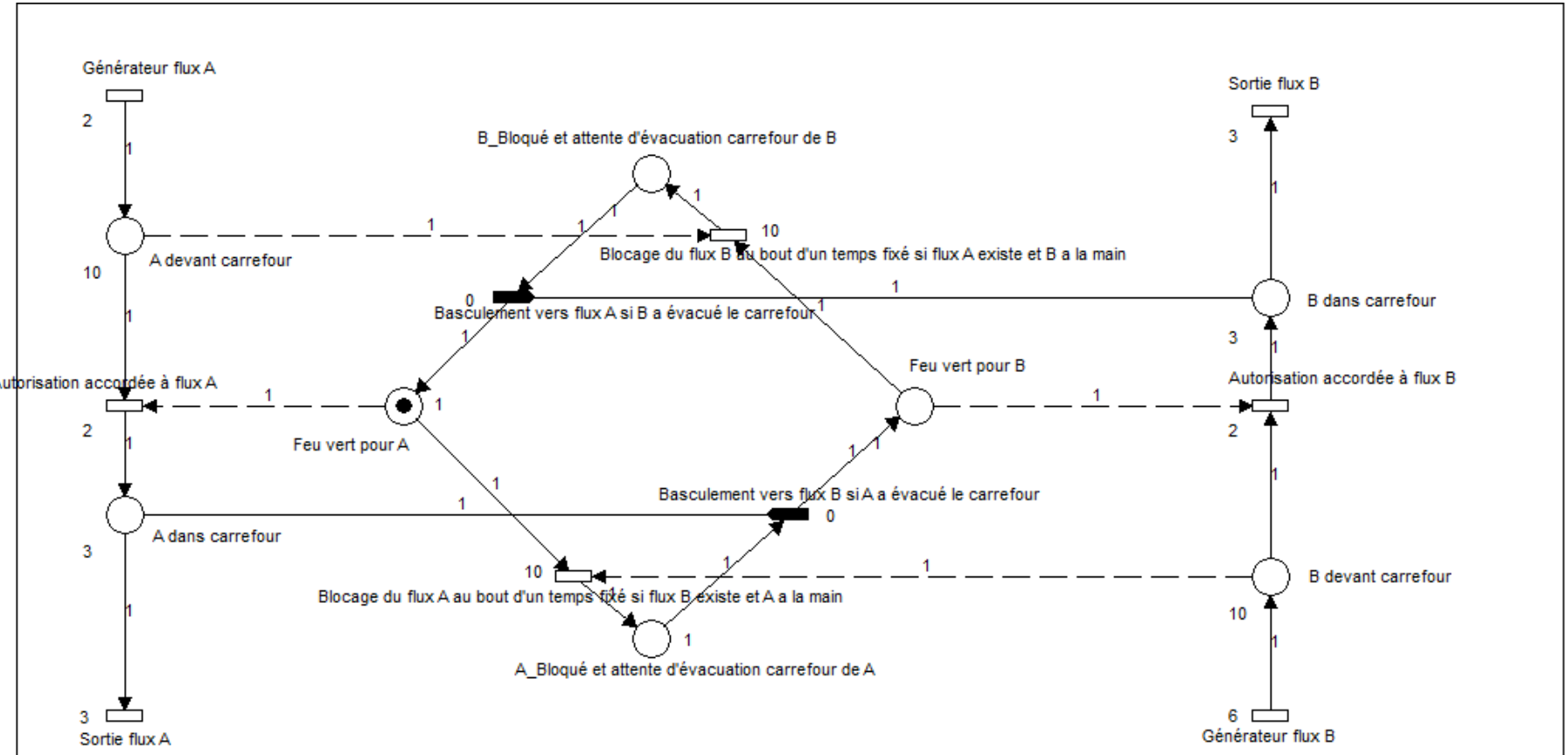


Les événements

- X_1 correspond à un arc de test provenant de la place Autorisation FA et Y_1 à un arc de test provenant de FB détecté devant le carrefour
- X_2 correspond à un arc de test provenant de la place Autorisation FB et Y_2 à un arc de test provenant de FA détecté devant le carrefour

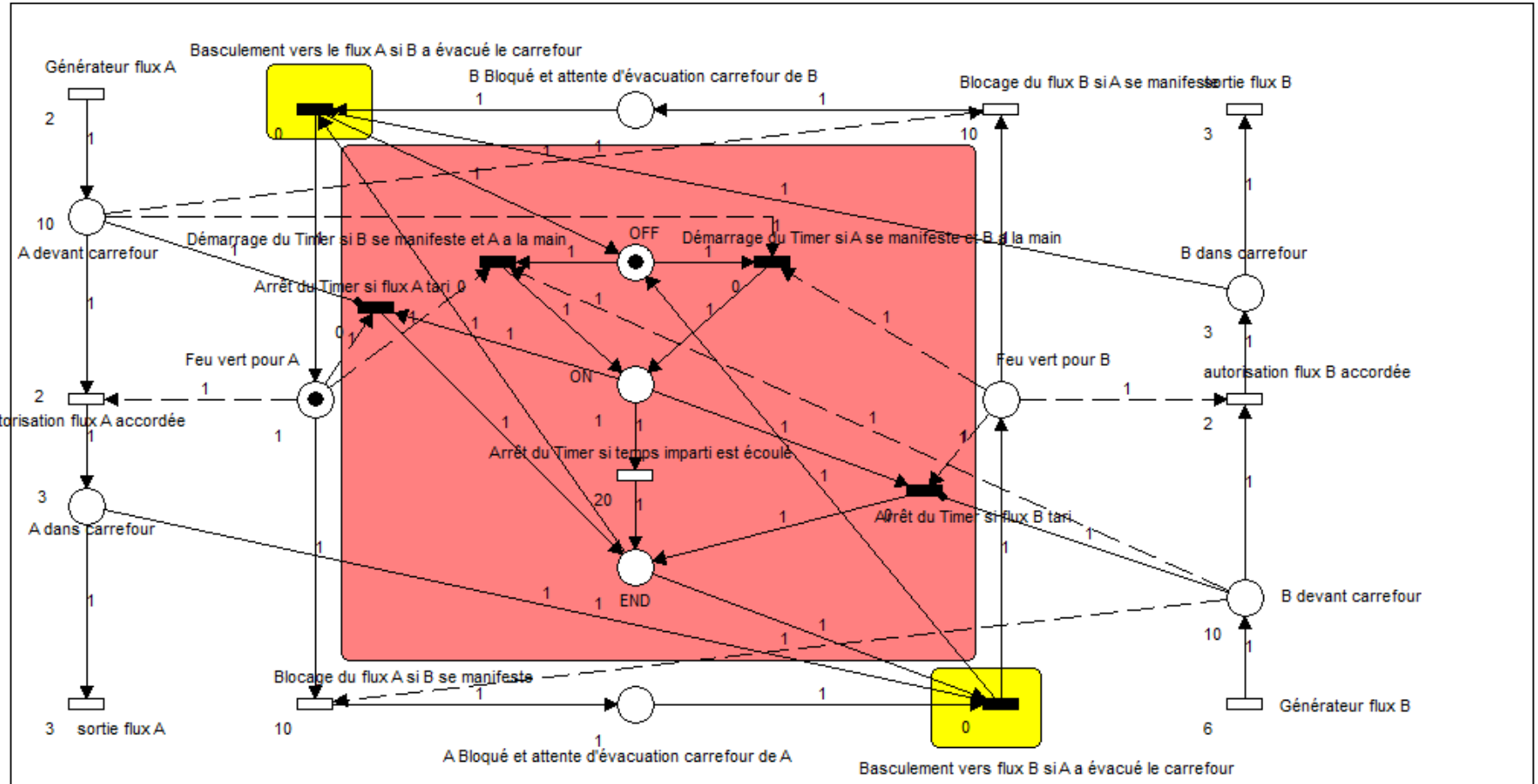
Système de gestion du carrefour – Version sans objet Timer

Version simple à implémenter avec un temps interne au RdP



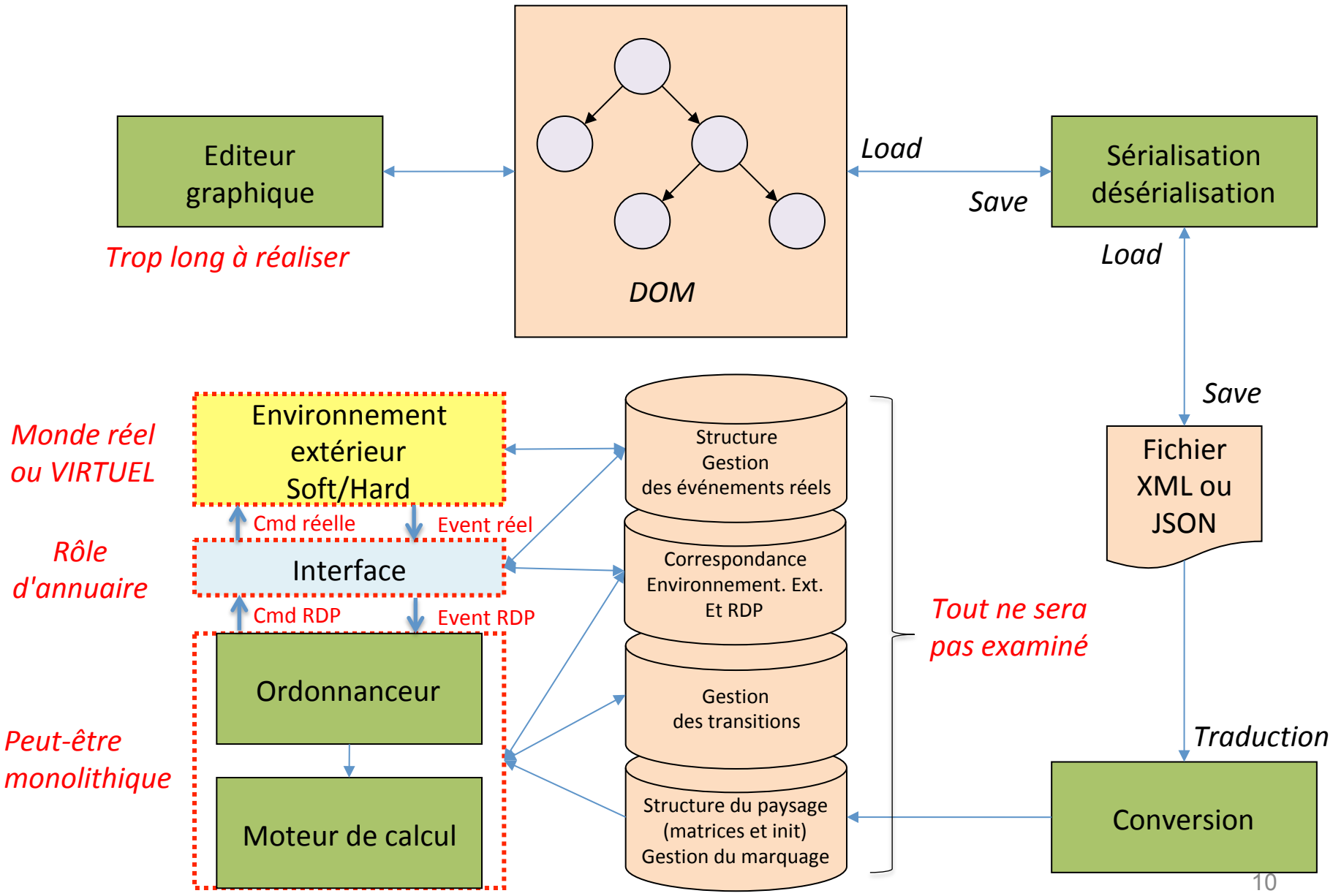
Système de gestion du carrefour – Version avec objet Timer

Version plus complexe à implémenter avec Timer (identifier l'amélioration à apporter)



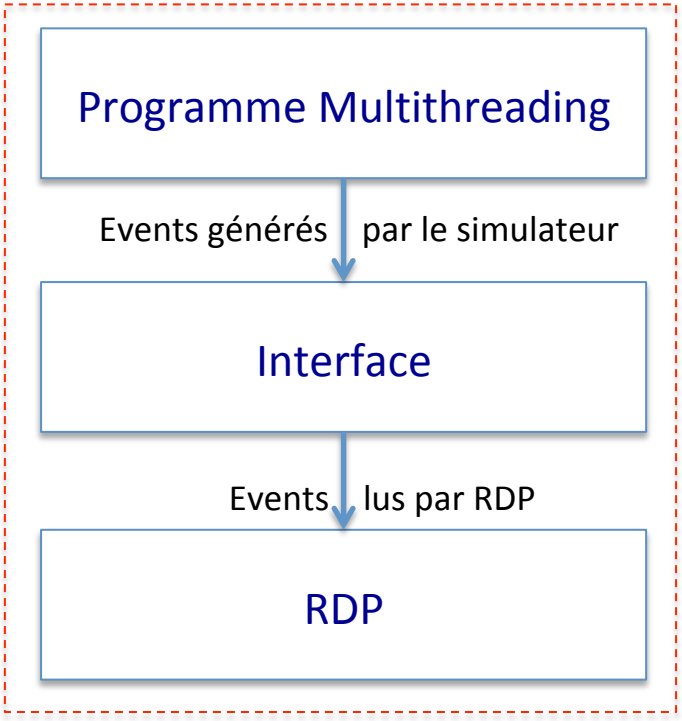
Fusion des transitions du système de gestion et du Timer

Architecture complète du système (pas le temps)



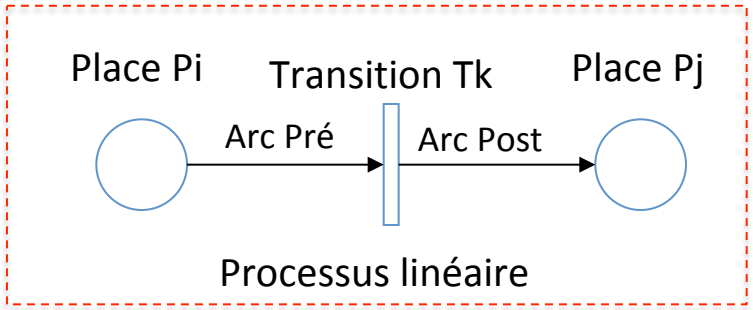
Architecture de base incluant le RdP

Solution proposée

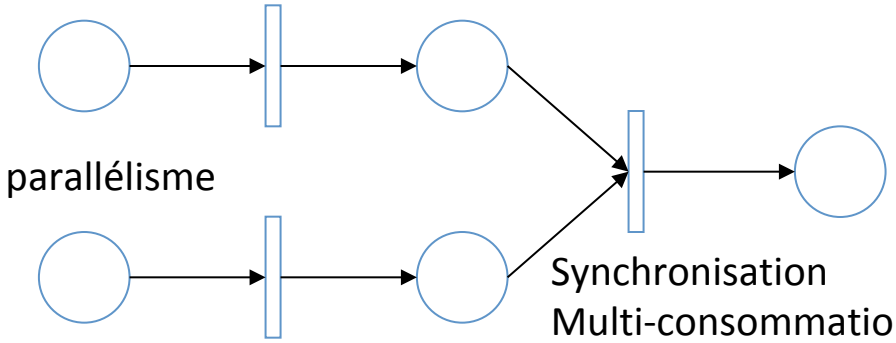


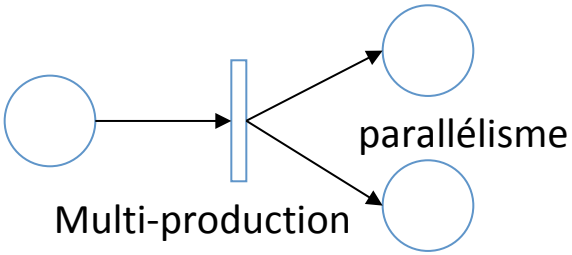
- Programme monolithique (tout est inclus)
- L'interface est une liste d'évènements notifiés par les acteurs du paysage

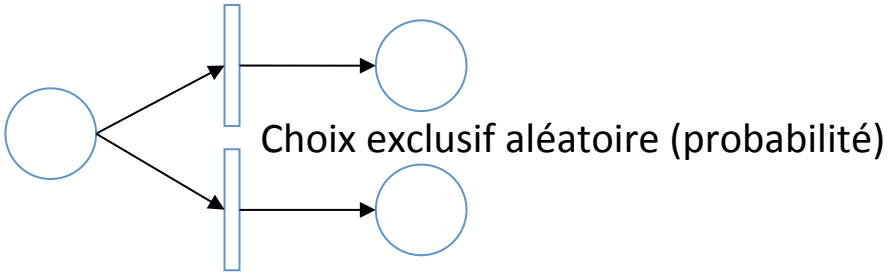
Le principe de base (rappel des définitions)



Minimum vital

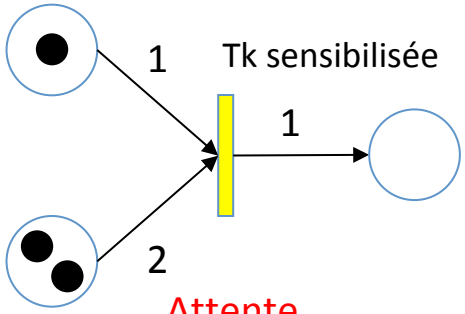




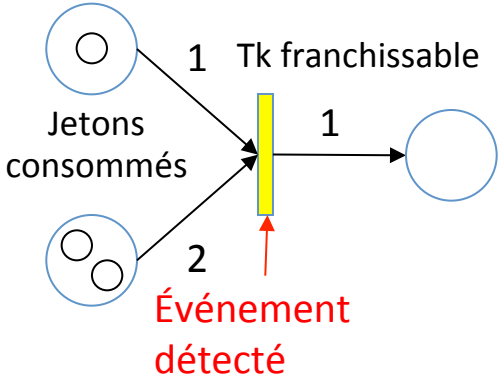


Gestion d'une transition en trois phases

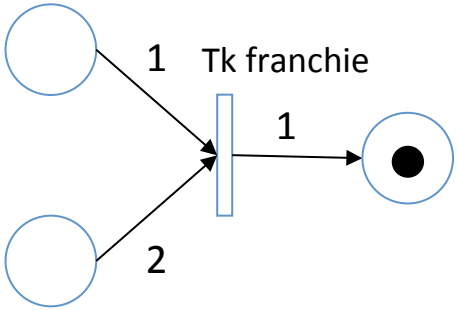
Φ1: Sensibilisation d'une transition



Φ2: Franchissement d'une transition

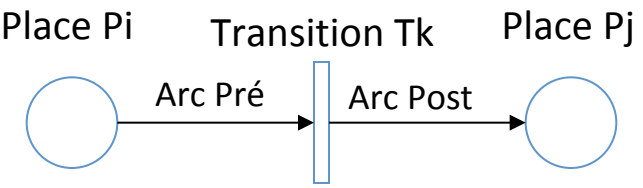


Φ3: production de jetons



Le principe de base (calcul matriciel)

Les éléments de base



P1	0
P2	1
P3	0
P4	0
P5	1
P6	2
P7	0

Marquage initial extrait depuis le fichier XML

	T1	T2	T3	T4	T5
P1	0	0	0	0	1
P2	1	0	0	0	0
P3	0	0	1	0	0
P4	0	2	0	0	0
P5	0	0	0	1	0
P6	0	1	0	0	0
P7	0	0	0	0	0

Matrice de pré-incidence extraite depuis le fichier XML (arcs reliant Places à Transitions)

	T1	T2	T3	T4	T5
P1	0	1	0	0	0
P2	0	0	0	1	0
P3	0	1	0	1	0
P4	0	0	0	1	0
P5	1	0	0	0	0
P6	0	0	1	0	2
P7	1	0	0	0	0

Matrice de post-incidence extraite depuis le fichier XML (arcs reliant Transitions à Places)

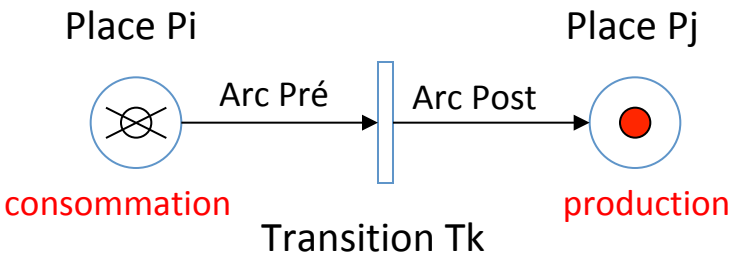
Remarques:

- Les matrices sont très creuses, ce qui justifie des structures de données adaptées (implémentation réseau, matrices, listes d'arcs,…)
- Les arcs doivent inclure (Pi, Tk, Poids) selon le sens (Pré ou Post), le poids donne la valeur de consommation ou de production

PAS LE TEMPS

Le principe de base (calcul matriciel)

L'évolution du marquage



P1	1	=	P1	0	-		T1	**T2**	T3	T4	T5	+		T1	**T2**	T3	T4	T5
P2	1		P2	1		P1	0	**0**	0	0	1		P1	0	**1**	0	0	0
P3	1		P3	0		P2	1	**0**	0	0	0		P2	0	**0**	0	1	0
P4	0		P4	2		P3	0	**0**	1	0	0		P3	0	**1**	0	1	0
P5	0		P5	0		P4	0	**2**	0	0	0		P4	0	**0**	0	1	0
P6	0		P6	1		P5	0	**0**	0	1	0		P5	1	**0**	0	0	0
P7	0		P7	0		P6	0	**1**	0	0	0		P6	0	**0**	1	0	2
						P7	0	**0**	0	0	0		P7	1	**0**	0	0	0

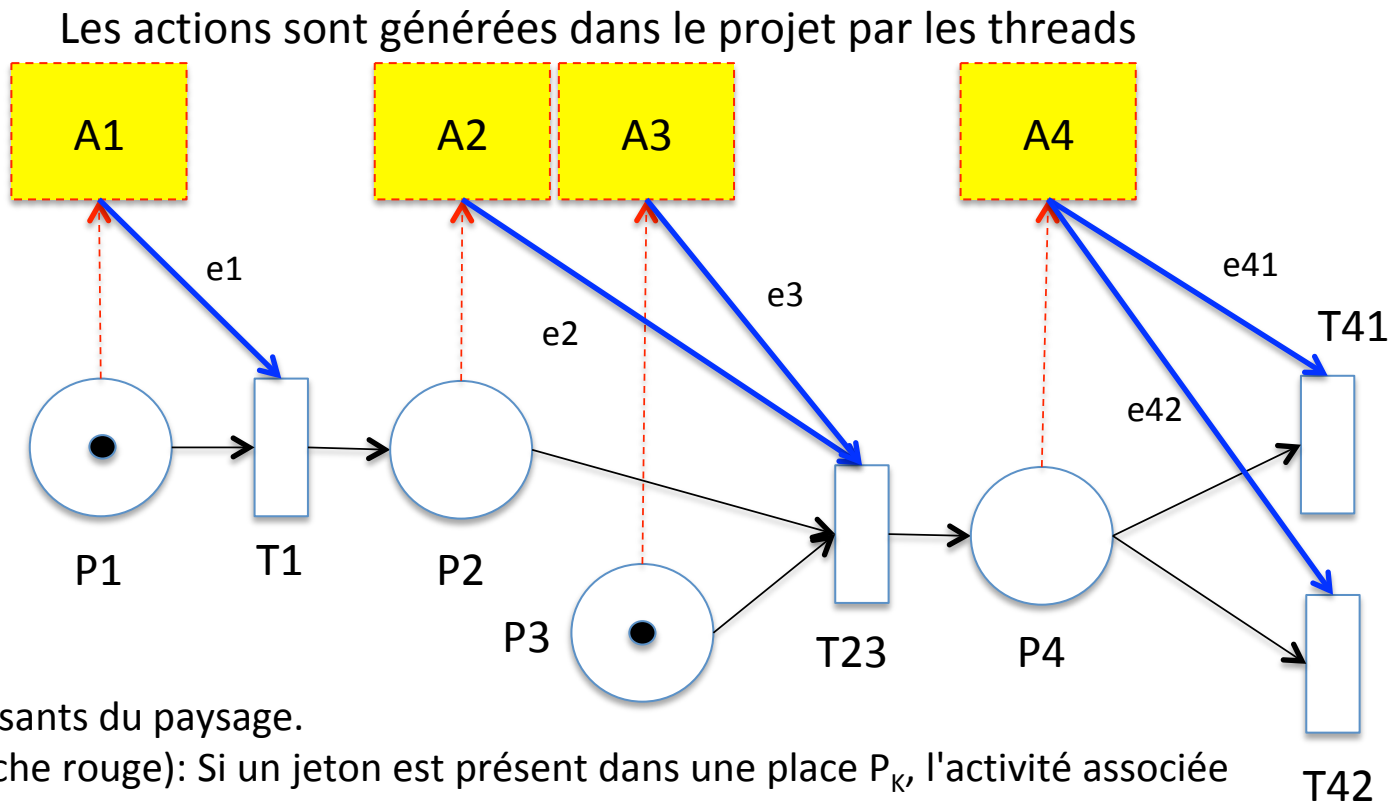
Nouveau Marquage — Ancien marquage — Consommation — Production

Remarques:

1. Le calcul aura effectivement lieu lorsque l'évènement associé à la transition aura été détecté
2. Dans cet exemple, on admet que la transition **T2** est sensibilisée. Si d'autres transitions étaient sensibilisées en même temps, il faudrait faire intervenir les colonnes associées aux transitions depuis les deux matrices. Ordre important (voir suite)

Gestion des événements (modèle à simplifier)

Le paysage ci-contre met en évidence des activités basiques et indépendantes:
A1, A2, A3 et A4, contrôlées par processus modélisé par un RDP



Interprétation des composants du paysage.

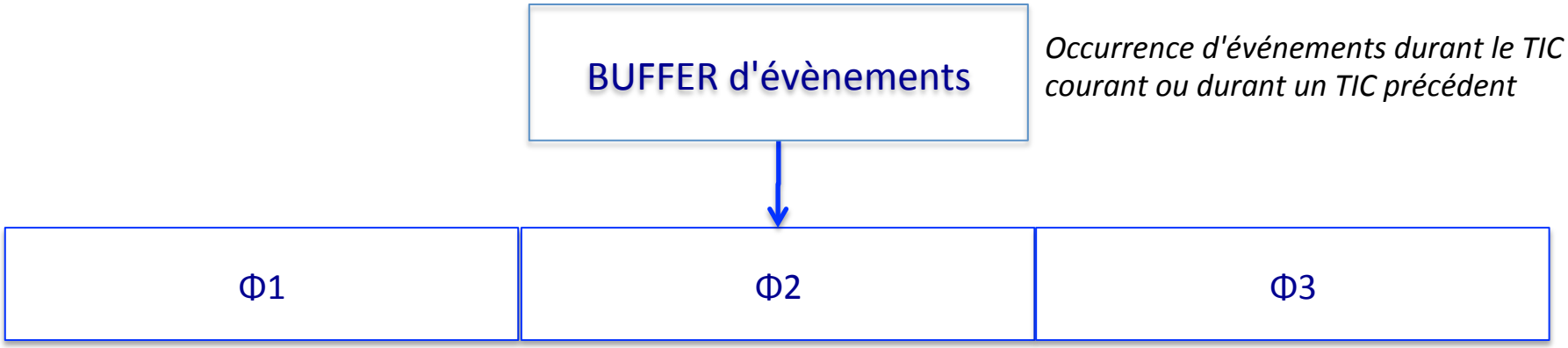
- Lancement action (flèche rouge): Si un jeton est présent dans une place P_K, l'activité associée A_K est lancé.
- Retour événement (flèche bleue): L'activité A1 retourne un événement mais A4 peut retourner deux événements possibles disjoints. Ces événements sont associés aux transitions (e1 issu de A1 est associé à T1, e2 et e3 issus respectivement de A2 et A3 sont associés à T2 (sous forme de ET logique,…). E41 ou e42 sont retournés par A4 et associés à T41 et T42

SOLUTION: comme pour la PRE et POST incidences, il faut une matrice d'association

Existe une phase Φ0
Lancement activités

Implémentation du réseau de Petri
La base de temps du processus (le TIC)

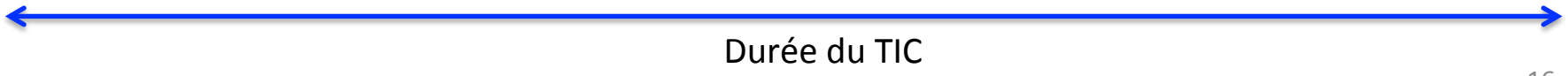
- Le TIC est la fenêtre temporelle dans laquelle le RDP peut évoluer.
- Durée du TIC dépend de la nature du processus (Temps réel ou pas)
- L'ensemble des événements se produisant durant le TIC est assimilable à une collision (instantanéité)



- Marquage intermédiaire
- Évaluation sur le réseau COMPLET de la sensibilisation des transitions en fonction du marquage

- Évaluation du franchissement des transitions
- Consommation de jetons avec modification des propriétés (?)
- Mise à jour du marquage
- Mise à jour des transitions sensibilisées depuis l'état du réseau PARTIEL

- Production de jetons avec modification des propriétés (?)
- Mise à jour du marquage
- Mise à jour des transitions franchissables
- Comparer marquage Φ2 et Φ3 et lancer activités



Correspondance activités et marquage d'une place [1]

- A chaque place correspond une ou plusieurs activités dont la complexité et le nombre dépendent de l'interprétation du marquage (pour l'instant RDP ordinaires ou généralisés)
- Le scheduler sait qu'une action doit être déclenchée et l'interface sait laquelle

Cas simple (RDP ordinaire)

Trois situations:

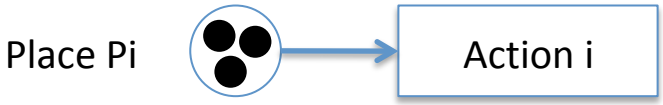
- Place non marquée -> pas d'action
- Place nouvellement marquée -> lancement action
- Place déjà marquée -> pas d'action

ATTENTION: cas particulier (consommation de jeton et injection d'un nouveau jeton) => deux phases

Cas moyen (RDP généralisé)

Cumul de jetons possible. Chaque jeton correspond au lancement de l'action correspondant à la place i
ATTENTION au même cas que précédemment

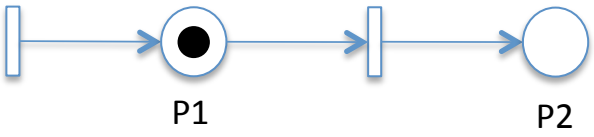
Cas complexe (non normalisé)



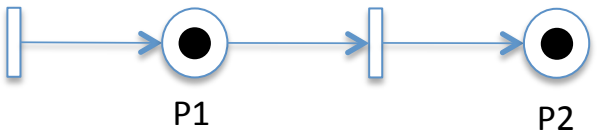
L'action est déclenchée sur une combinaison de jetons dont le nombre peut être fixe ou variable (changement de contexte)
ATTENTION au même cas que précédemment

Correspondance activités et marquage d'une place [2]
Effet de bord

Avant traitement

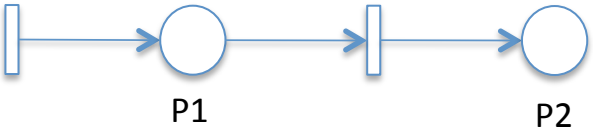


Après traitement



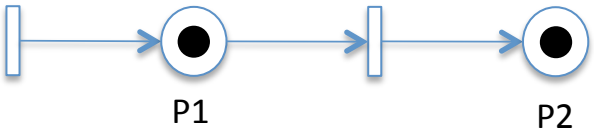
Une seule action déclenchée au lieu de 2
Alors que jeton injecté en P1 et en P2

Après Pré-incidence



Le jeton de P1 a été consommé

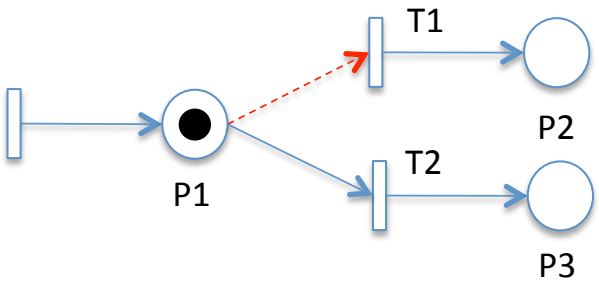
Après Post-incidence



Deux jetons ont été injectés

Il faut comparer la pré-incidence et la post-incidence

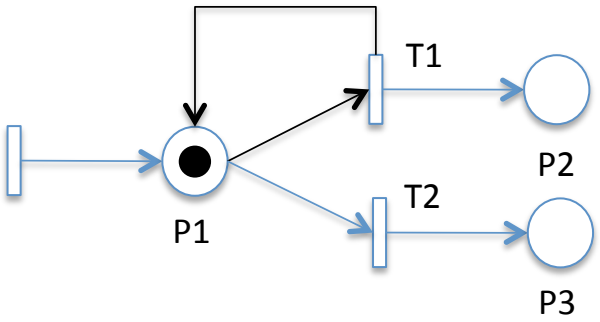
Correspondance activités et marquage d'une place [3]
Que se passe-t-il avec les arcs de test?



Si l'arc de test détecte la présence du jeton, on peut être confronté à trois cas:

- On attend un événement pour franchir la transition (sans consommation). L'événement étant traité, le test ne sera plus validé car place n'évolue plus et donc plus d'événement généré.
- Un événement peut ne pas exister, auquel cas on générerait en continu des jetons en P2 car T1 toujours sensibilisée (forcément une situation non souhaitée!).
- Si malgré tout on veut générer un jeton en P2 mais après un temps ΔT, il faudrait que l'action déclenche un Timer. Or l'action associée à P1 n'est déclenchée que lors de l'injection du jeton en P1 (donc qu'une seule fois)

Solution possible



Remarque: une place peut déclencher plusieurs actions)

Mécanismes
Algorithme simplifié du Scheduler

Itération globale à chaque Tick. Séparation du marquage courant en deux phases
franchissement des transitions doit être mémorisé: LISTEFRANCHIES nécessaire
La liste des événements est produite par l'extérieur LISTEEVENTS

Tant que (processus actif) faire {

Φ0: Pour chaque Place Pi du RDP faire {
si MarquagePOST(Pi) > MarquagePRE(Pi) alors lancer activité(Ai)
}

MarquagePRE = MarquagePOST //nouveau marquage courant

Φ1: Pour chaque Transition Tj du RDP vérifier si sensibilisée = f(PRE, marquagePRE)) et mettre dans liste LISTESENSIBILISEES

Ranger dans ordre aléatoire LISTESENSIBILISEES

Φ2: Pour chaque Tj ⊂ LISTESENSIBILISEES faire {

Si ET(LISTEEVENTS(Tj)) alors {

MarquagePRE = MarquagePRE – PRE(Ti, *)

extraire Ti de LISTESENSIBILISEES et mettre Ti dans LISTEFRANCHIES

Mettre à jour LISTESENSIBILISEES

//certaines transitions peuvent ne plus être sensibilisées

}

}

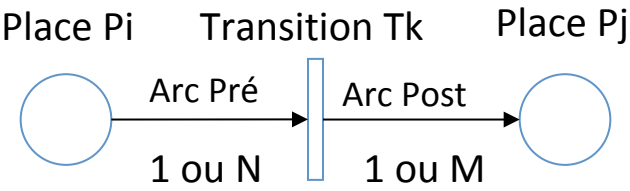
Φ3: MarquagePOST = MarquagePRE //nouveau marquage courant intermédiaire

extraire Ti de LISTEFRANCHIES et MarquagePOST = MarquagePOST + POST(Ti, *)

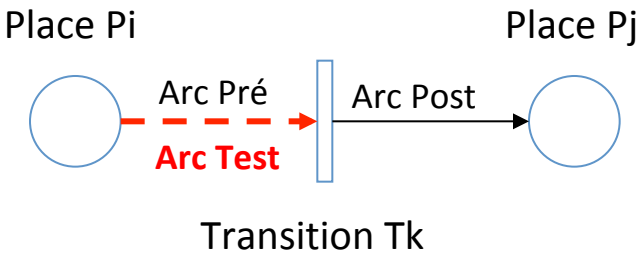
}

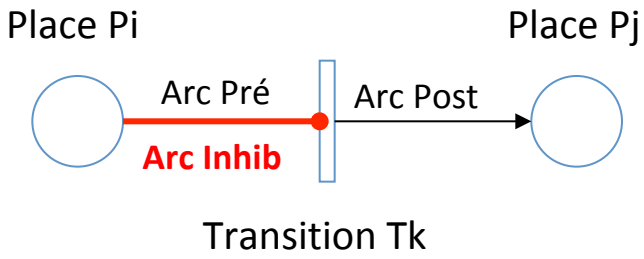
Il existe plusieurs types d'arcs

Les arcs normaux ordinaires et généralisés



nous devons aussi gérer des arcs de test et des arcs inhibiteurs.
Ces deux types d'arcs sont exploités uniquement en pré-incidence.





La gestion est plus complexe et affecte:

- l'ordonnanceur pour évaluer si une transition est sensibilisée
- le moteur de calcul car le test ne consomme rien (au même titre que l'arc inhibiteur)

Cette gestion nécessite une information supplémentaire pour le scheduler

- Arc (Pi, Tk, Poids, Nature)

En revanche, ne consommant rien, la matrice de pré-incidence inclura des valeurs 0 sur les arcs de tests ou inhibiteurs

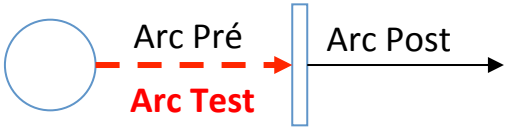
Plus tard, nous généraliserons les arcs de test (incluant inhibiteurs)

Nature évoluée des arcs de test

Dans HPSIM on distingue les arcs inhibiteurs et arcs de test.
Le test est validé si le poids de la place est plus grand ou égal au poids de l'arc

Possibilité de généraliser les arcs avec une palette complète de tests incluant inhibiteur:

- < strictement plus petit que
- <= plus petit ou égal
- > Strictement plus grand
- >= plus grand ou égal (concept implémenté dans HPSIM)
- != diifférent
- == égal (concept implémenté dans HPSIM pour les arcs inhibiteurs)



Transition Tk

La gestion est à peine plus complexe que la précédente:
Cette gestion nécessite un codage de la nature adapté 0 pour arc normal 1 à 6 pour les codes de tests

- Arc (Pi, Tk, Poids, Nature)

PAS LE TEMPS

Définition du projet [1]
Les solutions pour instancier le paysage

- Pas de graphisme donc aucune information de formes et de positions, uniquement les relations logiques.
- Coder en dur les matrices (très long, risque d'erreurs, difficile à faire évoluer, chaque application nécessite de recompiler)
- Fichier texte simple sans les événements à discuter ultérieurement

```
//Commentaire NB de places. La suite de places est décrite avec la valeur initiale du nombre de
    jetons et évetuellement la capacité
N
P1 init capacity
…..
//Commentaire NB de transitions
M
T1 T2 T3 …..
//Commentaire NB d'arces de pré-incidence. Chaque arc doit décrire la place et la transition
    associée, son poids et sa nature. Chaque information séparée par des espaces
K1
Pi Tj Wij Nature
…..
//Commentaire NB d'arces de post-incidence. Chaque arc doit décrire la place et la transition
    associée, son poids. Sa nature n'a pas de sens ici. Chaque information séparée par des espaces
K2
Pk Tl Wkl
…..
```

PAS LE TEMPS

Définition du projet [2]
Création d'un fichier paysage à parser

Exploitation d'un document texte ou XML

- Proposition d'un format de type XML (nettement plus riche mais nécessite un parser XML)

```
<paysage>
    <places>
        <place name="…" capacity="…" ID="…" initToken="…">définition</place>
    </places>
    <transitions>
        <transition name="…" ID="…">définition</transition>
    </transitions>
    <arcsPRE>
        <arc IDREFplace="…"  IDREFtransition="…" Poids="…"  Nature="…"/>
    </arcsPRE>
    <arcsPOST>
        <arc IDREFplace="…"  IDREFtransition="…" Poids="…"/>
    </arcsPOST>
</paysage>
```