Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Centre Universitaire de Mila
Laboratoire MISC, Université Constantine 2–Abdelhamid Mehri

Cours Ingénierie des logicielles

Chapitre 03

Transformation de Graphes

Présenté par:

Dr Said Meghzili

2021

Plan du chapitre

- Introduction
- Concepts de Transformation de Graphe
  - Principes de L'IDM
  - Transformation de modèle
  - Transformation de graphe
  - L'outil AToMPM
- Transformation des BPMN vers RdP
  - Les formalismes source BPMN et cible Rdp
  - Méta-modélisation
  - Règles de transformation
- Exemple de Transformation d'un modèle BPMN
- Conclusion

Introduction

Ce chapitre Contient deux Partie :

1) La première partie concerne les concepts théorique :

- ✓**Rappelle sur IDM : Ingénierie Dirigée par les Modèles, Méta-modélisation et transformation de modèles.**

- ✓ **Notion de graphe , transformation de graphes, réécriture de graphe, grammaire de graphe, outils de transformation de graphes.**

2) La deuxièmes partie est une étude de cas: transformation des modèles BPMN vers RdP en utilisant AToMPM:

- ✓**Méta-modélisation des modèles BPMN et RdP**
- ✓**Règles de passage de BPMN vers RdP**
- ✓**Exemple De transformation d'un modèle source**

Principes de L'IDM :Ingénierie Dirigée par les Modèles

- ✓ **Approche de développement mettant à disposition de l'utilisateur des concepts, des langages et des outils**

- ✓ **Les modèles sont considérés comme des éléments de base**

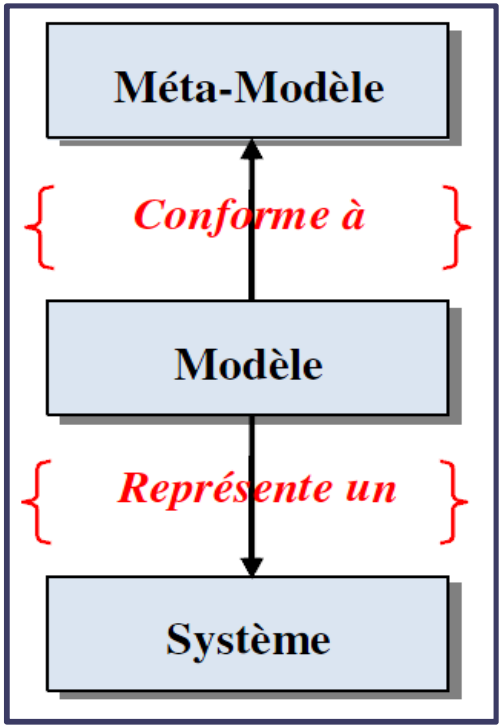
- ✓ **Raisonnement est entièrement à un haut niveau d'abstraction**

- ✓ **L'application sera générée (en tout ou en partie, automatiquement ou semi-automatiquement) à partir de modèles**

- ✓ **Les outils permettant de créer et d'exploiter ces modèles sont construits autour des concepts de: Méta-modélisation, Transformation de modèles**

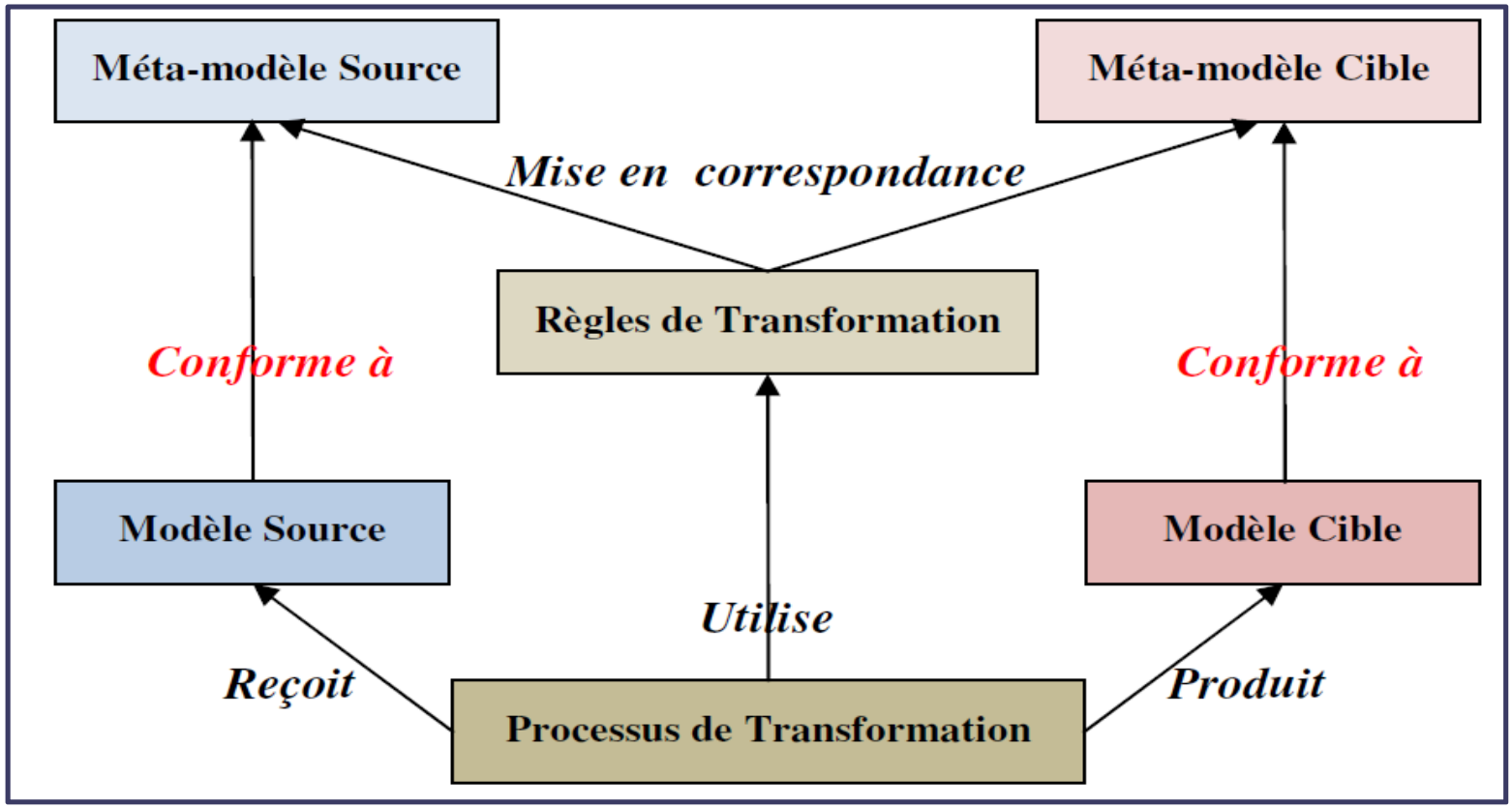
Principes de L'IDM : Méta-modélisation

- Modèles
- Méta-modèles



Relation entre système, modèle et méta-modèle

Principes de L'IDM : Transformation de modèles



Concepts de base de la transformation de modèles

Transformation de graphes

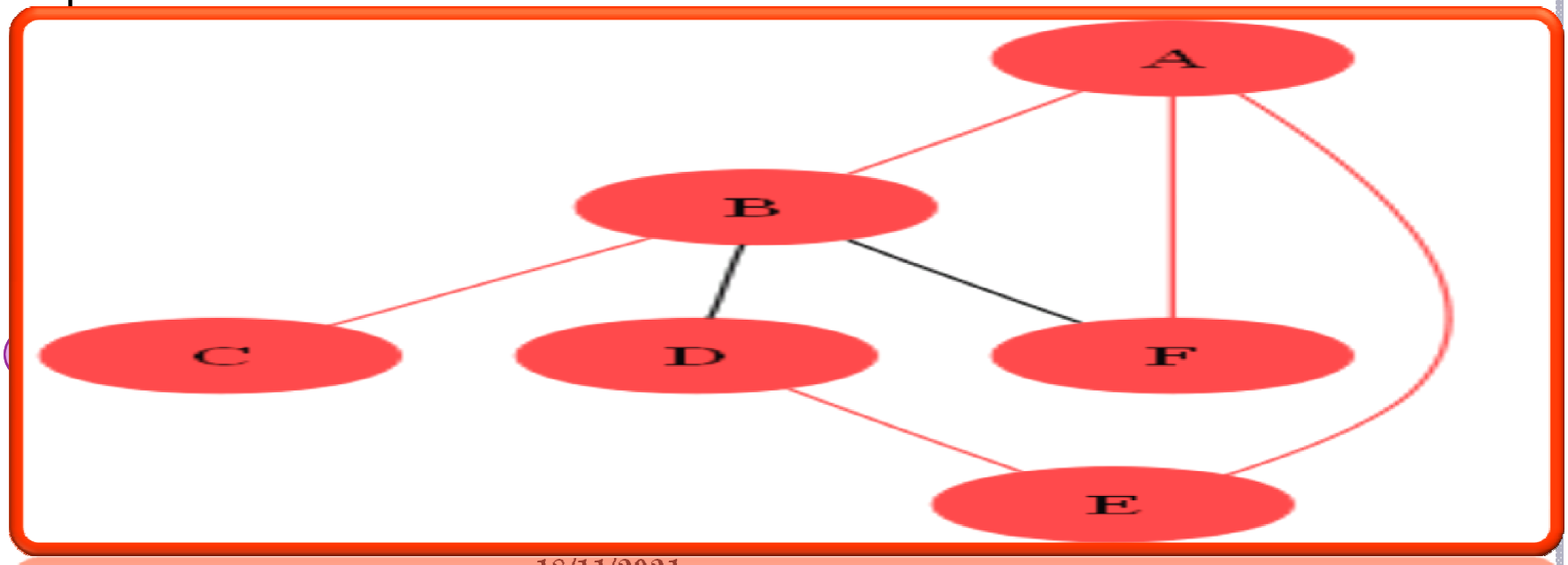
A. Concept de graphe

1. Le Graphe

Un graphe est constitué de sommets ou nœuds qui sont reliés par des arêtes. Plus formellement, on appelle graphe G = {S, A}

- **S** : ensemble fini non vide d'éléments appelés sommets ou nœuds.
- **A** : ensemble fini non vide de paires de sommets appelés arcs.
- $A \subseteq X \times X = \{(s, t) | s, t \in S\}$

Chaque arc de **A** relie deux sommets de S.

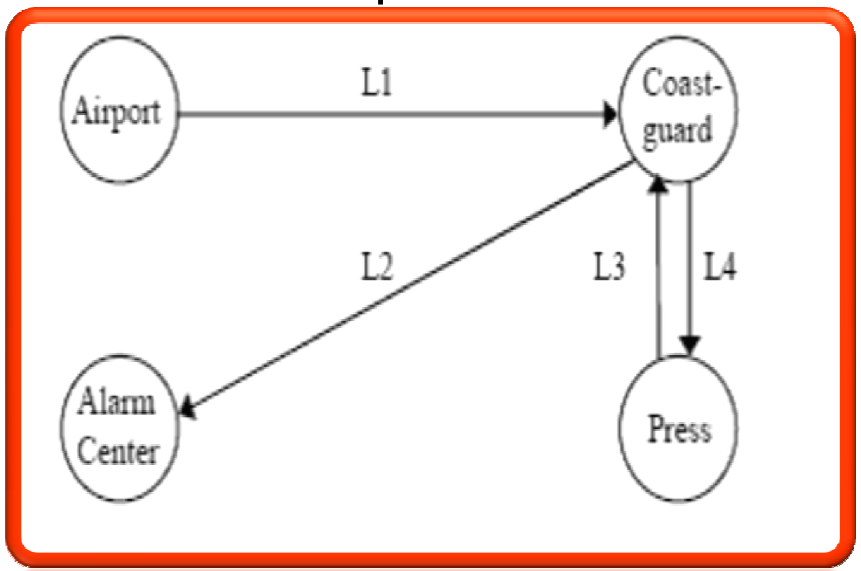


Transformation de graphes

A. Concept de graphe

4. Graphe étiqueté:

Un graphe étiqueté est un graphe orienté dans lequel les arcs possèdent un ensemble non vide d'étiquettes.



5. Le Degré d'un graphe:

le degré d'un sommet dans un graphe est le nombre d'arêtes dans ce sommet. Si le graphe est orienté le degré de sommet se définit en degré entrant et degré sortant, relatifs respectivement au nombre d'arcs entrant et sortants dans ce sommet.

Transformation de graphes

A. Concept de graphe

6. L'ordre d'un graphe :

l'ordre d'un graphe G se définit par le nombre de sommets présents dans ce graphe .

7. Le Chemin dans un graphe:

le chemin dans un graphe se définit par la succession des arcs parcourus dans un sens définit dans le graphe, il possède les propriétés et les caractéristiques suivantes :

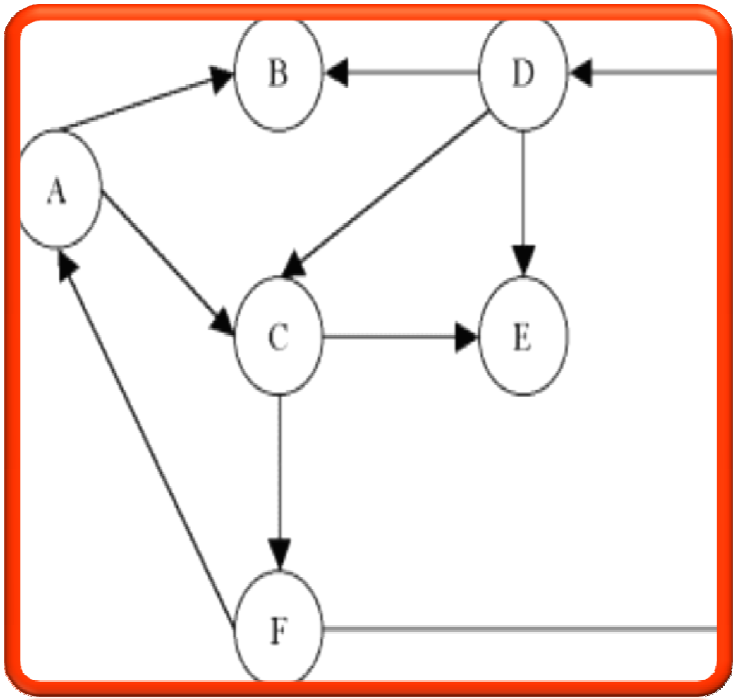
- ✓ la longueur du chemin est égale au nombre d'arcs parcourus .
- ✓ le chenin est dit une chaîne si l'on ne tient pas compte de la direction des arcs .
- ✓ le chemin est dit circuit s'il revient à son point de départ .
- ✓ la distance entre deux sommets dans un graphe se définit comme étant la longueur du plus court chemin entre ces deux sommets, si cette longueur est de 1 les sommets sont dits adjacents.
- ✓ le diamètre d'un graphe est la plus grande distance séparant deux sommets dans ce graphe.

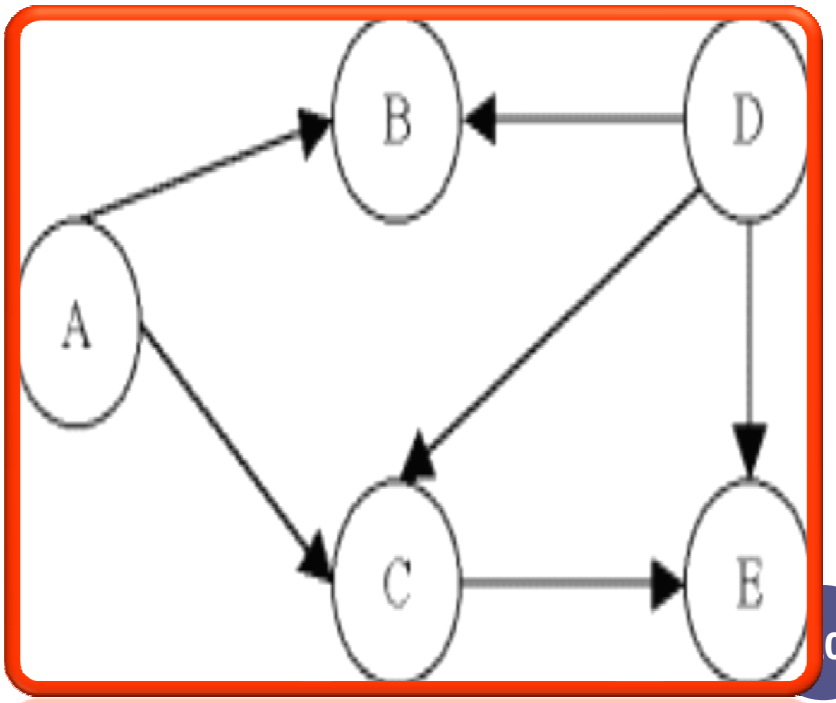
Transformation de graphes

A. Concept de graphe

8. Le sous-graphe:

un sous graphe G'(S',A') d'un graphe G(S,A) est un graphe composé d'un sous ensemble de sommet S'∈ S et d'un sous ensemble d'arêtes A'∈A tel que A' représente les arêtes reliant les sommets S' dans le graphe G'





Transformation de graphes

B. Système de transformation de graphes

1. Principes des règles

Une règle de transformation de graphe est définie par : r = (L, R, K, glue, emb, cond)
Elle consiste en:

- Deux graphes **L** graphe de côté gauche et **R g**raphe de côté droit.
- Un sous graphe **K** de **L.**
- Une occurrence g**lue** de **K** dans **R** qui relie le sous graphe avec le graphe de côté droit.
- **emb,** Une relation d'enfoncement qui relie les sommets du graphe de côté gauche et ceux du graphe du côté droit.
- Un ensemble **cond** qui spécifie les conditions d'application de la règle.

2. L'application des règles

L'application d'une règle r= (L, R, K, glue, emb, cond) à un graphe G produit un graphe résultant H. Le graphe H fourni, peut être obtenu depuis le graphe d'origine G en passant par les cinq étapes suivantes :

A. Choisir une occurrence du graphe de côté gauche L dans G.

B. Vérifier les conditions d'application d'après cond.

C. Retirer l'occurrence de L (jusqu'à K) de G ainsi que les arcs pendillé, c.-à-d. tous les arcs qui ont perdu leurs sources et/ou leurs destinations. Ce qui fournit le graphe de contexte D de L qui a laissé une occurrence de K.

B. Système de transformation de graphes

D. Coller le graphe de contexte D et le graphe de côté droit R suivant l'occurrence de K dans D et dans R. c'est la construction de l'union de disjonction de D et R et, pour chaque point dans K, identifier le point correspondant dans D avec le point correspondant dans R.

E. Enfoncer le graphe du côté droit dans le graphe de contexte de L suivant la relation d'enfoncement emb .

L'application de r sur un graphe G pour fournir un graphe **H est appelée une dérivation directe depuis G vers H à travers r,** elle est dénotée par G ⇒ H ou simplement par G ⇒ H.

En donnant les notions de règle et de dérivation directe comme étant les concepts élémentaires de la transformation de graphe, on peut définir les systèmes de transformation de graphe, les grammaires de graphe et **la notion de langages engendrés**

3. Langage engendré

Soit G 0 un graphe initial, G n est le graphe final et une séquence de transformations successives de graphes : G 0 → G 1 → ... → G n est une dérivation successive à partir de G 0 vers G n en appliquant les règles de R qui sont étiquetées par les symboles de T, est dit langage engendré par R, G 0 et T et on écrit L(R, G 0 , T)

Transformation de graphes

Outils de transformation de graphes

Plusieurs outils de transformation de graphes existent actuellement, parmi lesquels :

- **AGG** [AGG]:The **A**ttributed **G**raph **G**rammar System
- **AToM3**:**A** **T**ool for **M**ulti-formalism and **M**eta-**M**odelling
- ***A ToMPM :*** *A Tool for Multi-Paradigm Modeling*
- **VIATRA** :**VI**sual **A**utomated model **TRA**nsformations.
- **FUJABA**: **F**rom **U**ML to **J**ava **a**nd **b**ack **a**gain.
- **GreAT** : The **G**raph **Re**write And **T**ransformation tool suite.

Transformation de graphes : l'outil AToMPM

- ✓ **AToMPM signifie « A Tool for Multi-Paradigm Modeling »**
- ✓ **Outil pour la modélisation multi-paradigme**
- ✓ **Successeur de AToM3**
- ✓ **Il basé complètement sur le Web**
- ✓ **Il fonctionne sur le nuage**
- ✓ **Effectuer des transformations de modèles, et manipuler et gérer des modèles**

Transformation de graphes : l'outil AToMPM

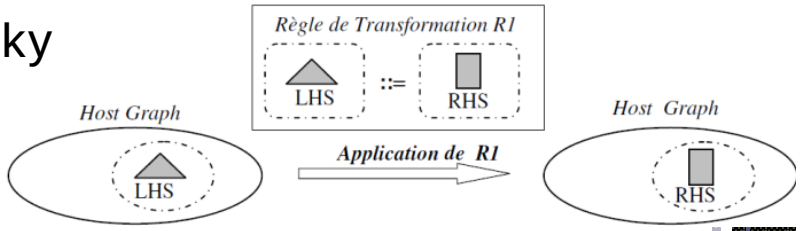
- AToMPM possède:
  - ➔ Une couche de Méta-modélisation
    - La modélisation graphique d'un formalisme
      - ***les entités du formalisme,***
      - ***leurs attributs,***
      - ***leurs relations,***
      - ***leurs cardinalités,***
      - ***leurs contraintes,***
      - ***leurs apparences graphiques (syntaxe concrète)***
    - La Génération automatique un outil pour manipuler les différents modèles dércits dans le formalisme spécifié
  - ➔ Un système de réécriture de graphes
    - Les manipulations de modèles par application itérative des règles d'une Grammaire de Graphes

Transformation de graphes : l'outil AToMPM

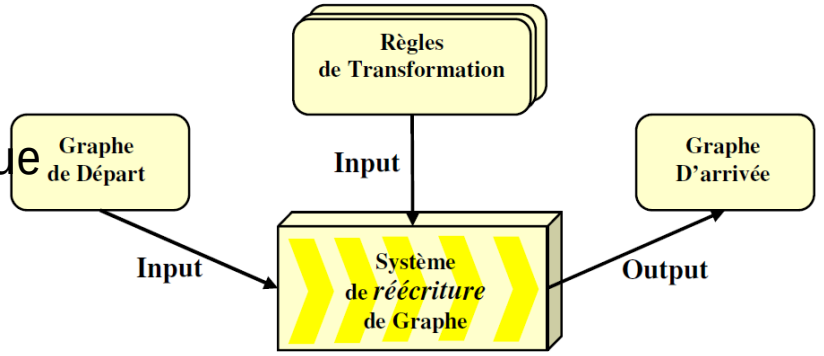
- Des techniques directement applicables à la transformation de modèles
- La transformation est:
  - Spécifiée sous forme d'un modèle de ***Grammaires de Graphes***
  - Effectuée par un ***Système de Réécriture de Graphes***
- Grammaires de Graphes
  - Généralisation des grammaires de Chomsky
  - Elles sont composées de règles
  - Une règle est constituée de deux parties:

    La partie gauche (LHS): destinée à être mise en correspondance

    La partie droite (RHS): décrit la modifica[illegible]



- Système de Réécriture de Graphes
  - Application des règles jusqu'à ce que aucune règle ne soit applicable



- ***Avantage:*** approche formelle fondée sur des bases mathématiques:
  - ✓ La théorie des graphes
  - ✓ Les grammaires formelles
  - ✓ La réécriture de termes

Transformation de graphes : *Grammaires de Graphes*

Un (méta-)* modèle est un graphe.

- Tout comme les grammaires de chaînes, mais les règles sont des graphes dans : LHS et RHS.
- Si une correspondance est trouvée entre un LHS et une zone dans le graphe, ce sous-graphe est remplacé par le RHS.
- Un système de réécriture de graphes essaie chaque règle dans l'ordre jusqu'à ce qu'aucune d'entre elles ne soit applicable.

Transformation de graphes : *Grammaires de Graphes*

Nous utilisons des grammaires de graphes pour :

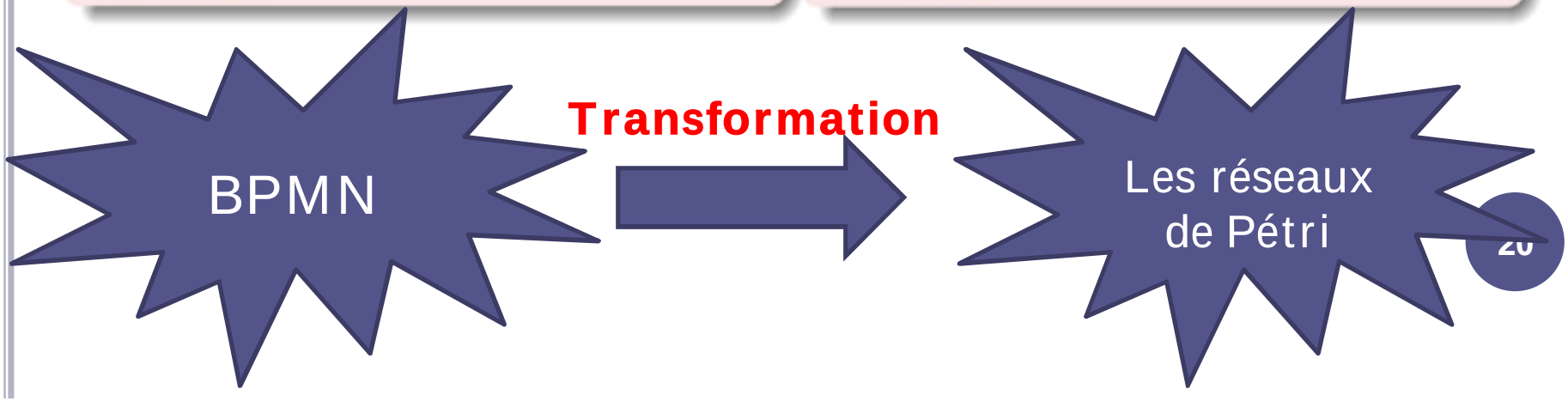
- Exprimer la sémantique opérationnelle (spécification du simulateur)
- Transformer des modèles en modèles comportementaux équivalents exprimés dans un autre formalisme.
- Optimiser les modèles.
- Générer du code pour un outil particulier (sémantique dénotationnelle).

Plan du chapitre

- Concepts de Transformation de Graphe
  - Principes de L'IDM
  - Transformation de modèle
  - Transformation de graphe
  - L'outil AToMPM
- **Transformation des BPMN vers RdP**
  - **Les formalismes source BPMN et cible Rdp**
  - **Méta-modélisation**
  - **Règles de transformation**
- Exemple de Transformation d'un modèle BPMN
- Conclusion

Problèmes & Motivations

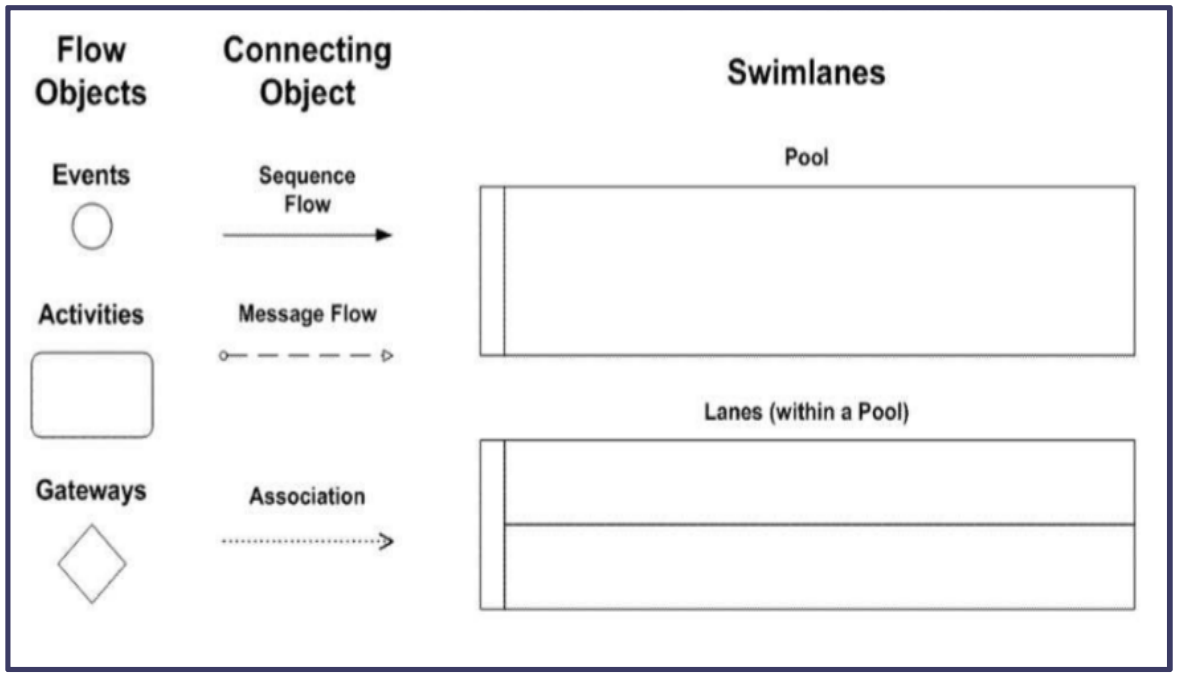
BPMN	PN
Avantages	**Avantages**
• Norme pour la modélisation de processus dans l'organisation • Notation graphique compréhensible • ... etc.	• Fondements mathématiques rigoureux • Plusieurs téchniques d'analyse et outils de vérification • ... etc.
Inconvénients	**Inconvénients**
• Manque de sémantique formelle • Analyse formelle de modèles impossible • ... etc.	• Notation complexe • Difficile à apprendre et à utiliser • ... etc.



Le formalisme source : Langage BPMN

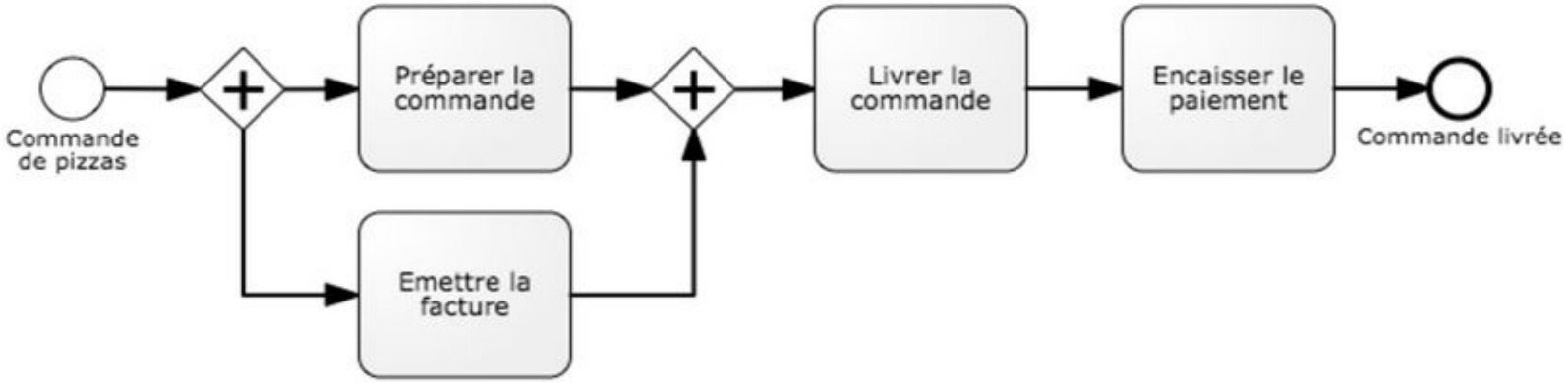
BPMN "Business Process Model and Notation" est un langage pour la modélisation des processus métier.

L'objectif principal du langage BPMN est de fournir une notation standard qui soit simple, facile et compréhensible par tous les utilisateurs métiers.



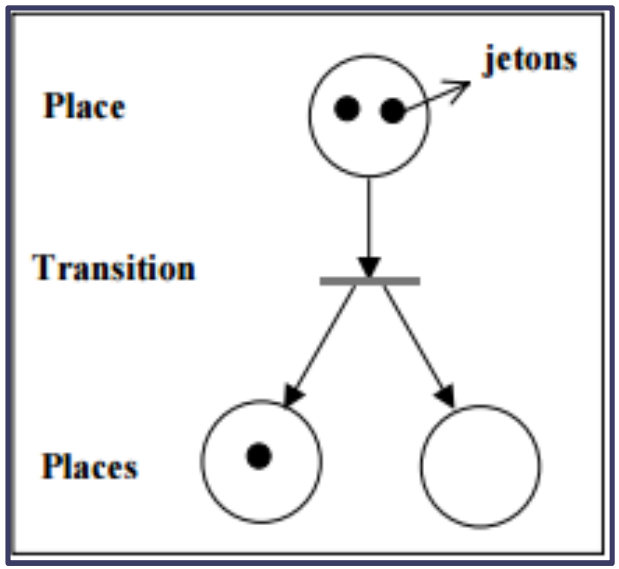
Le formalisme source : Langage BPMN

Exemple : la pizzeria



Le formalisme cible : Les réseaux de Pétri

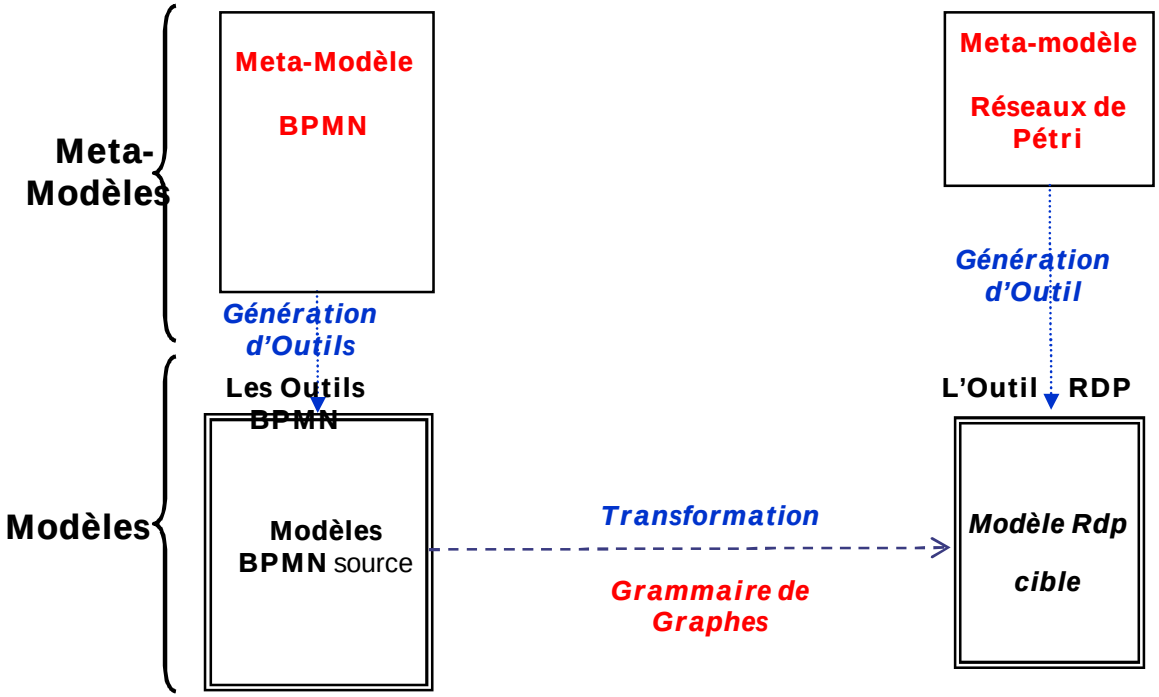
Un réseau de Pétri (aussi connu comme un réseau de Place/Transition) est un modèle graphique et mathématique permettant de modéliser et de vérifier le comportement dynamique des systèmes.



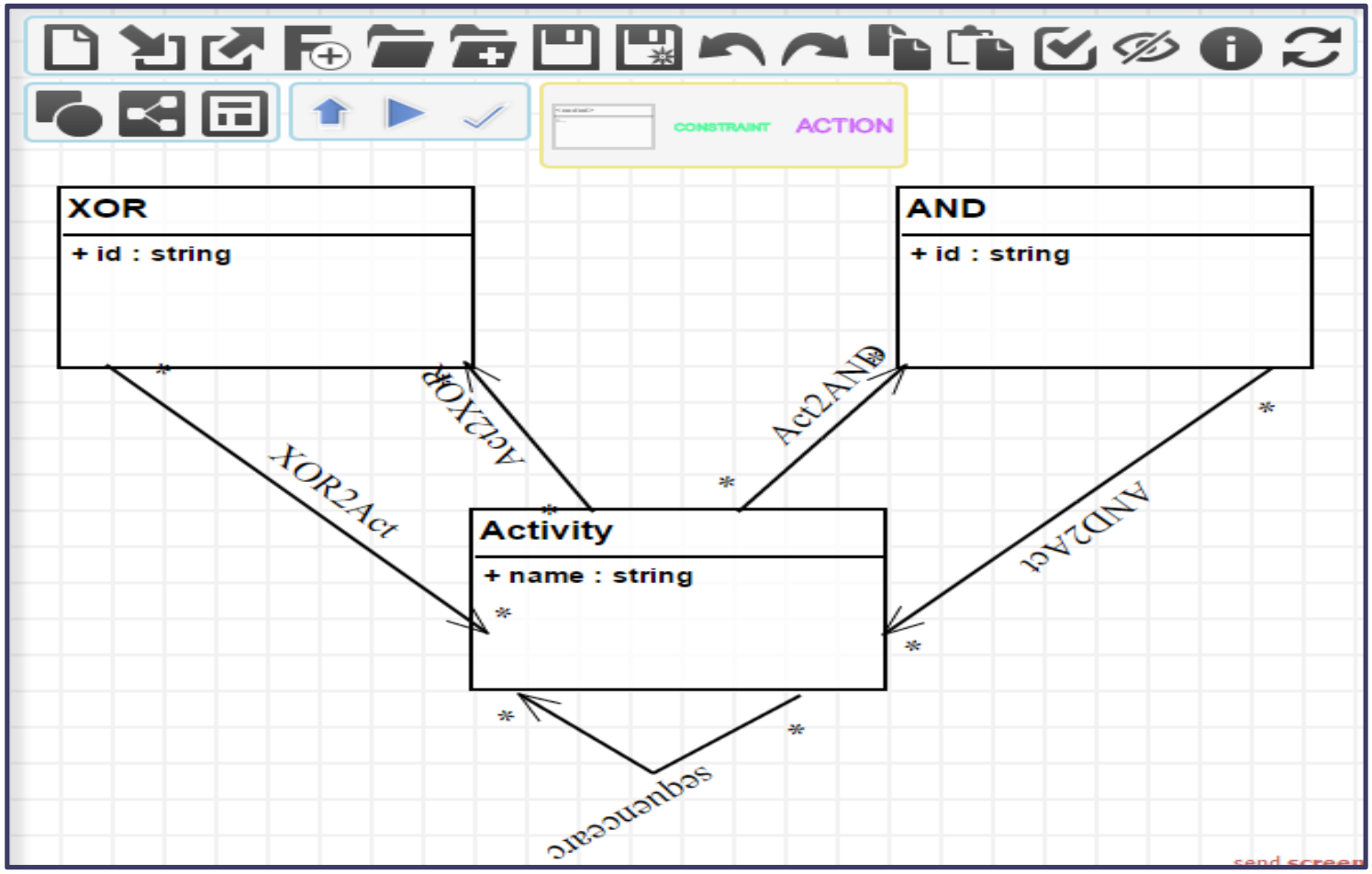
Architecture de la transformation: BPMN vers RdP

Pour transformer les modèles BPMN vers les réseaux de Pétri :

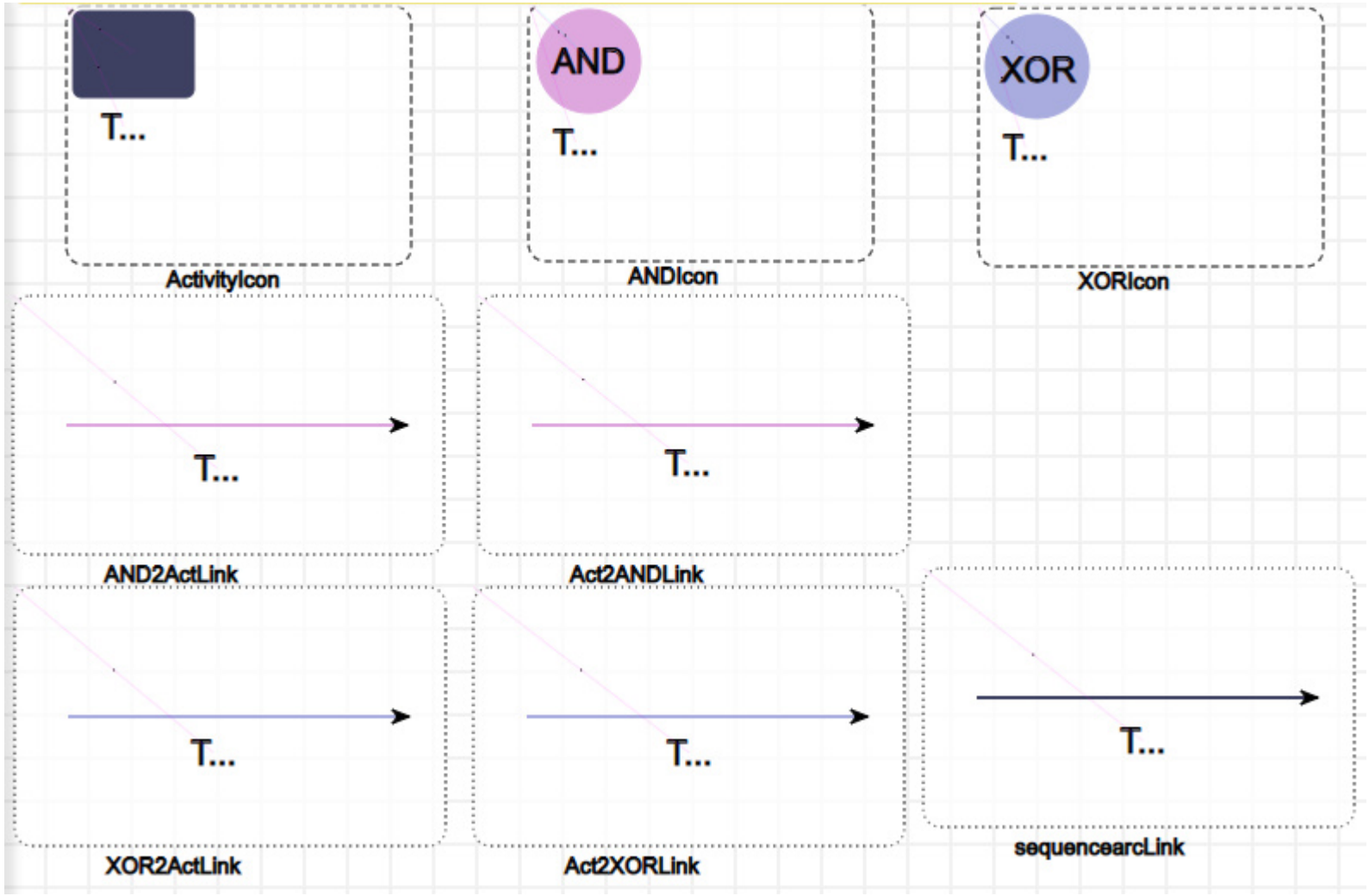
- **Construction d'un méta-modèle pour le langage BPMN**
- **Construction d'un méta-modèle pour les réseaux de Pétri**
- **Définition des règles de la transformation (Grammaire de Graphes)**



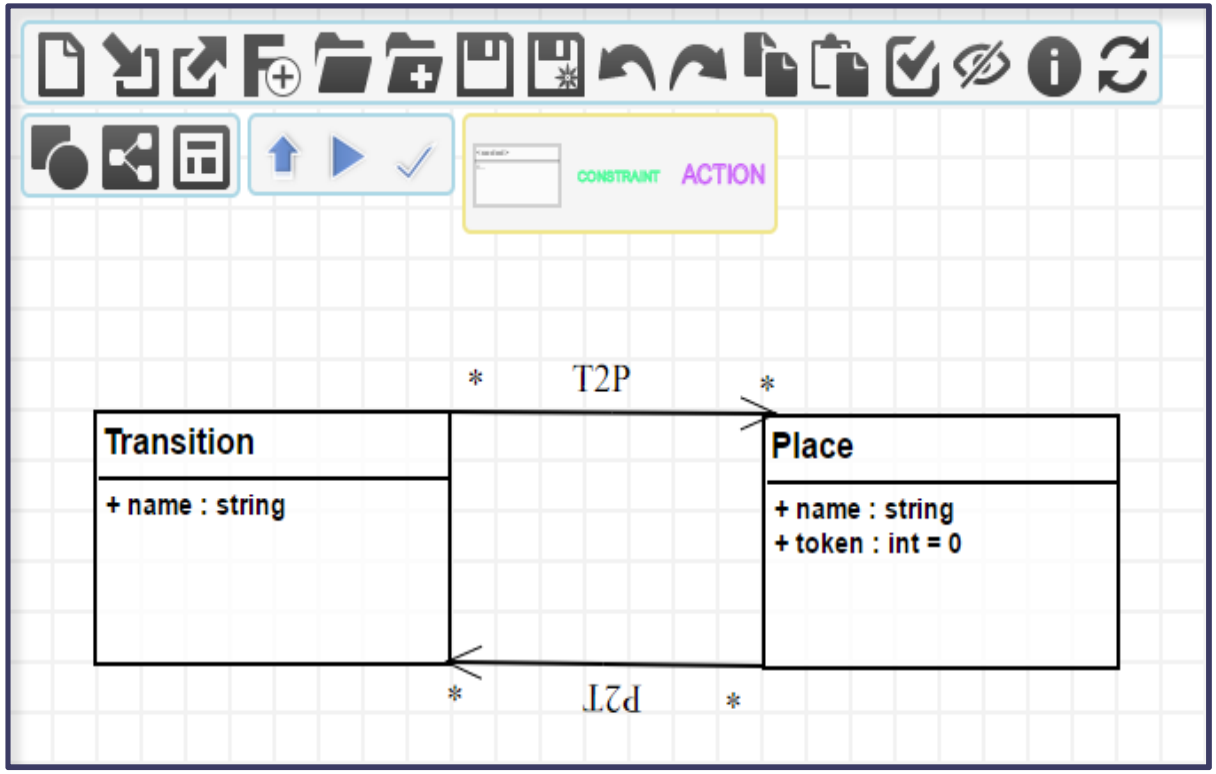
Méta-modèle du langage BPMN (Syntaxe Abstraite)



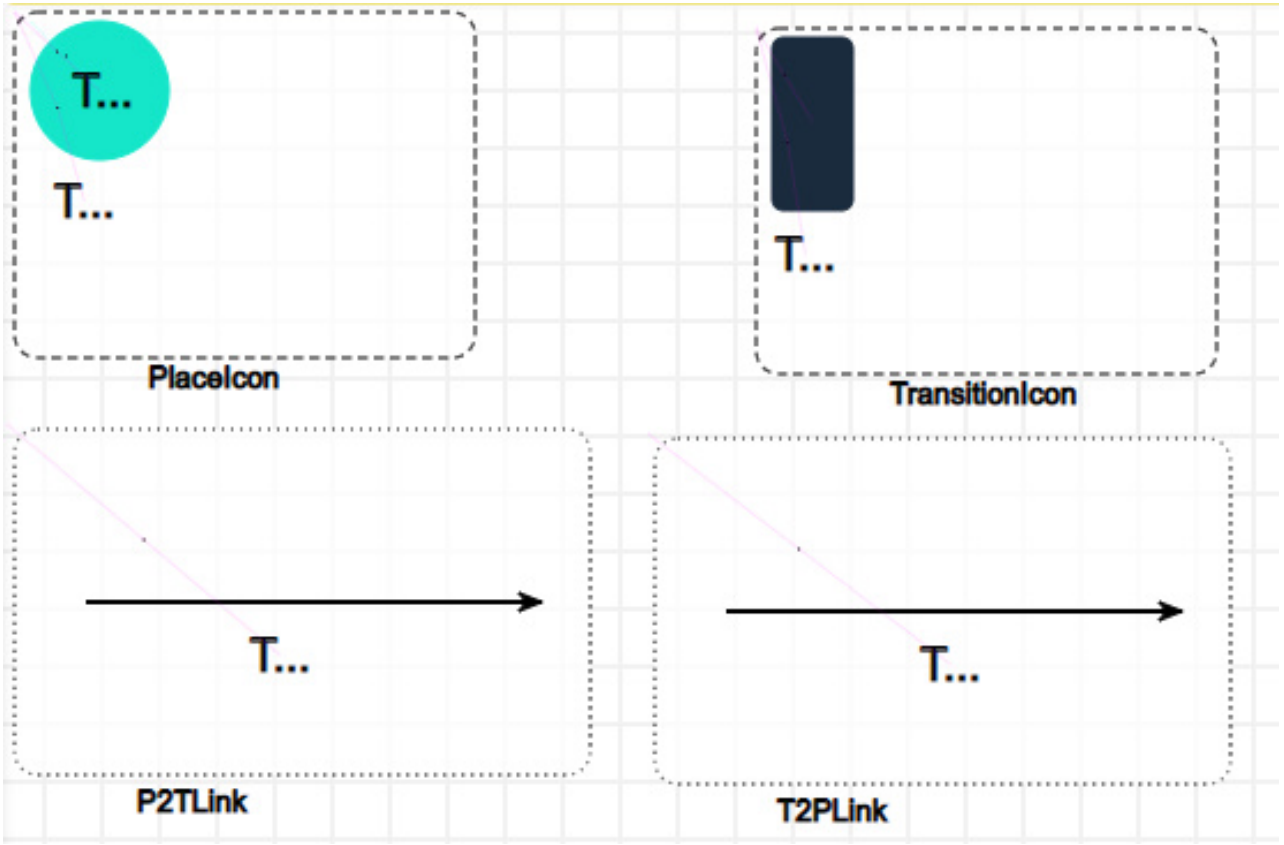
Syntaxe Concrète du langage BPMN



Méta-modèle des Réseaux de Pétri (Syntaxe Abstraite)



Syntaxe concrète des Réseaux de Pétri



Idée de la transformation BPMN vers RDP

L'idée de la transformation BPMN vers Rdp proposée :

- 1 les activités sont transformés en des transitions
- 2 les connecteurs XOR sont transformés en des places
- 3 les connecteurs AND sont transformés en des places

Règles de la transformation BPMN vers RDP

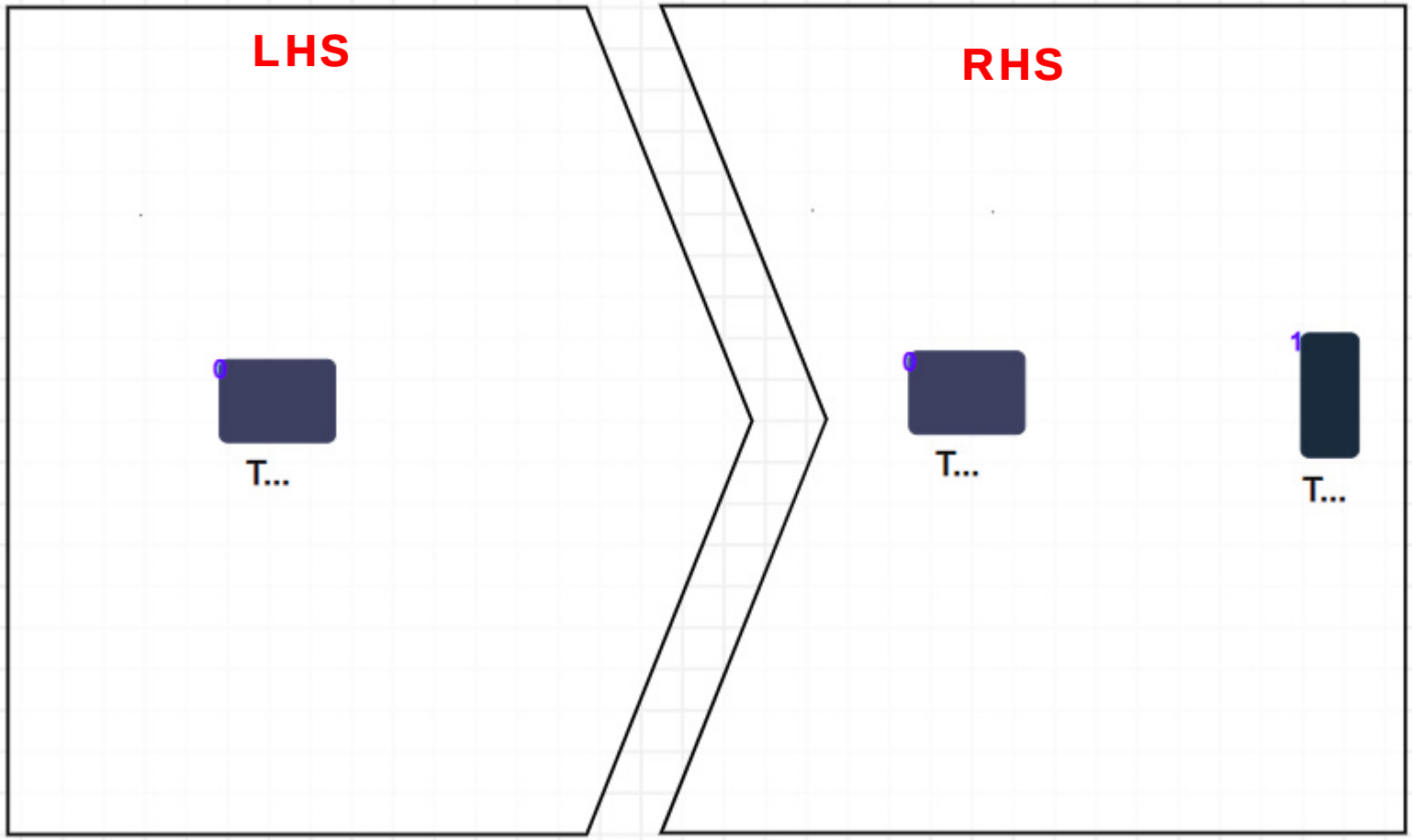
Les règles de la transformation BPMN vers Rdp proposée :

- Règle 1 : R_Activity2Transition
- Règle 2 : R_sequencearc
- Règle 3 : R_xor2place
- Règle 4 : R_inXOR2inPL
- Règle 5 : R_outXOR2outPL
- Règle 6 : R_ANDrule
- Règle 7: R_DeleteXOR
- Règle 8: R_DeleteAND
- Règle 9: R_DeleteActivity

Règles de la transformation BPMN vers RDP

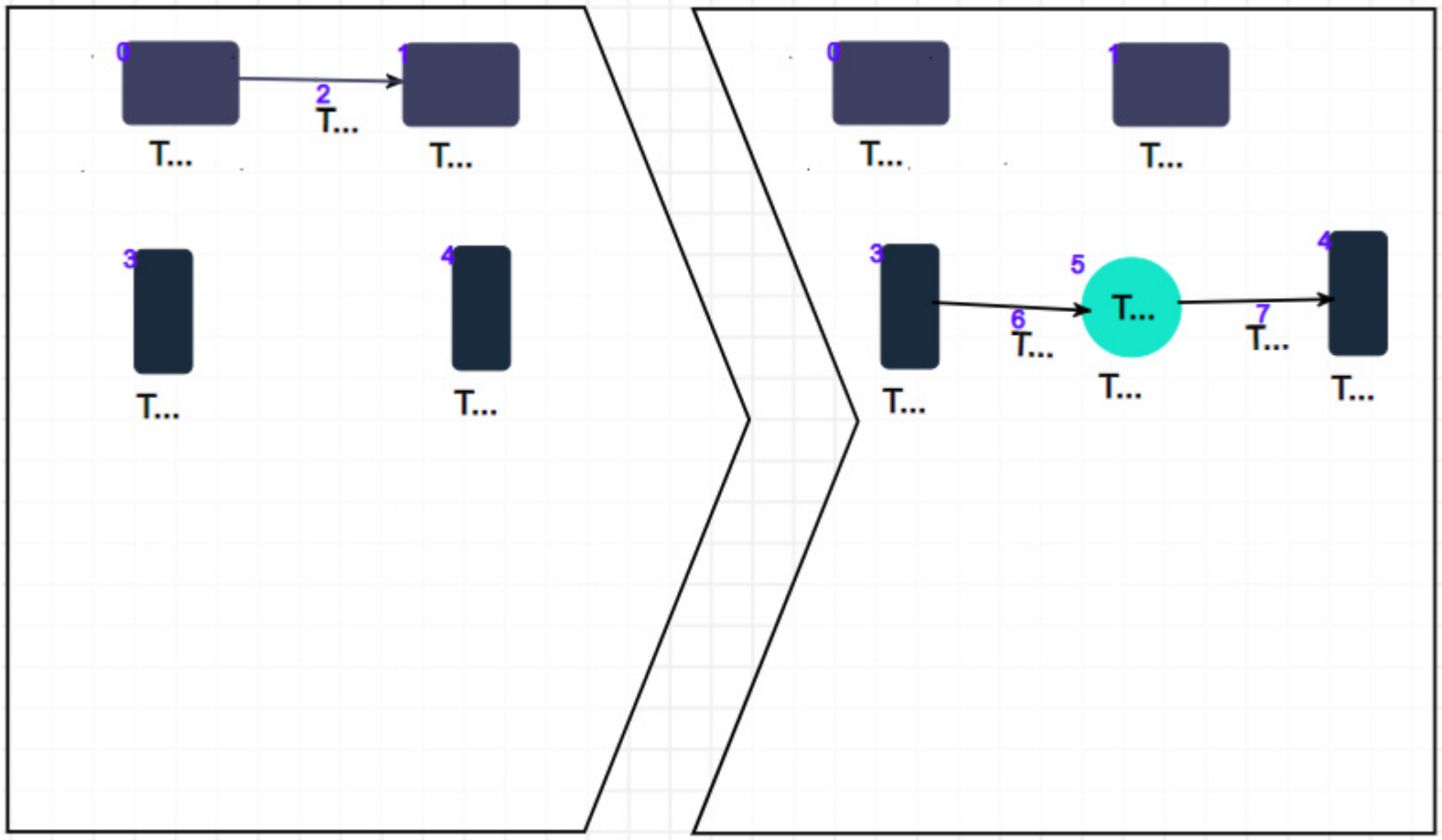
- **Règle 1 R_Activity2Transition :**

  Générer une nouvelle transition (dans RdP) pour chaque activité (dans BPMN). **LHS ::= RHS** (LHS sera remplacé Par RHS)



Règles de la transformation BPMN vers RDP

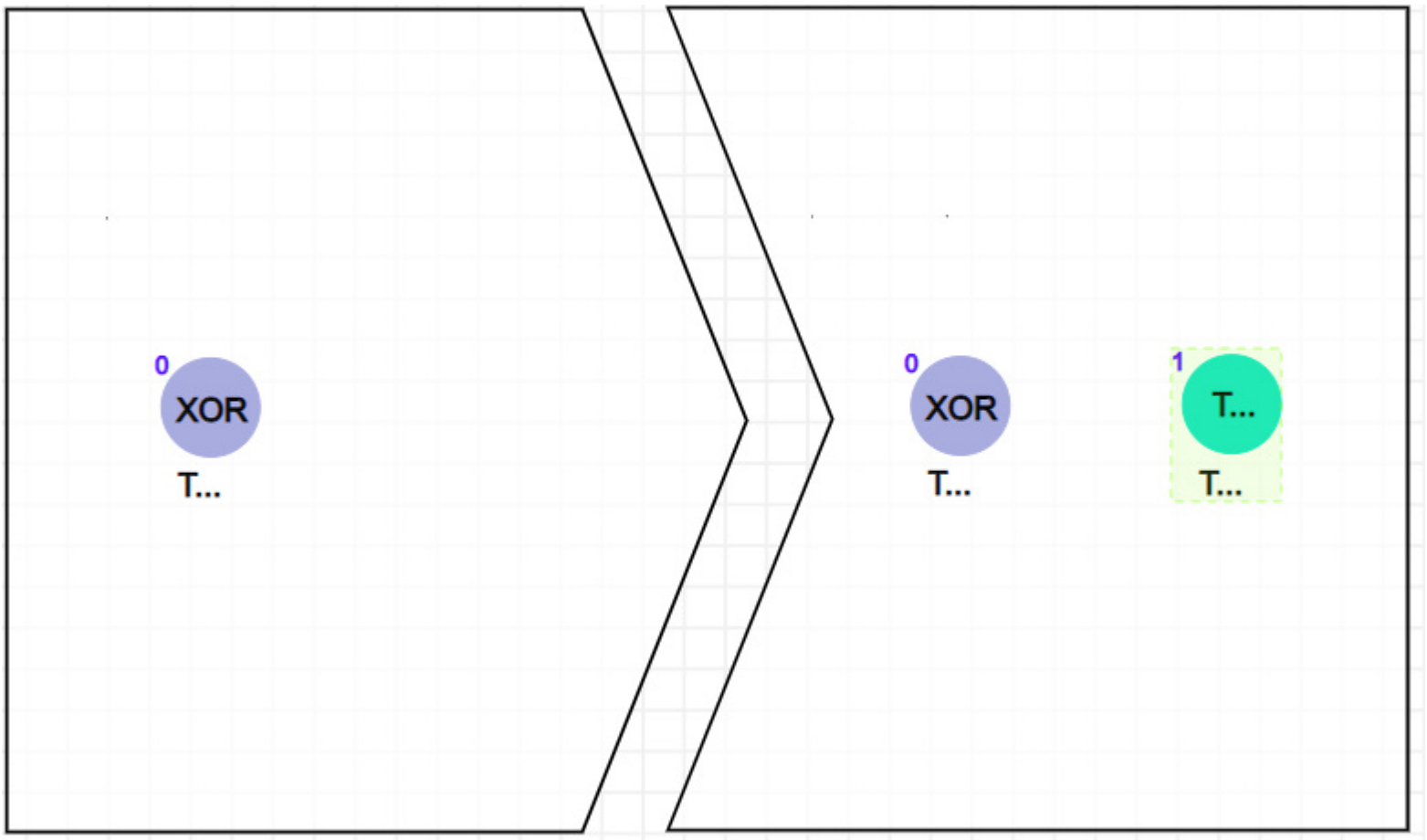
- **Règle 2 R_sequencearc:** Cette règle est appliquée à chaque arc de séquence:
  - •Génère une nouvelle place avec un arc entrant et un arc sortant aux transitions.
  - •Supprime tous les arcs de séquence.



Règles de la transformation BPMN vers RDP

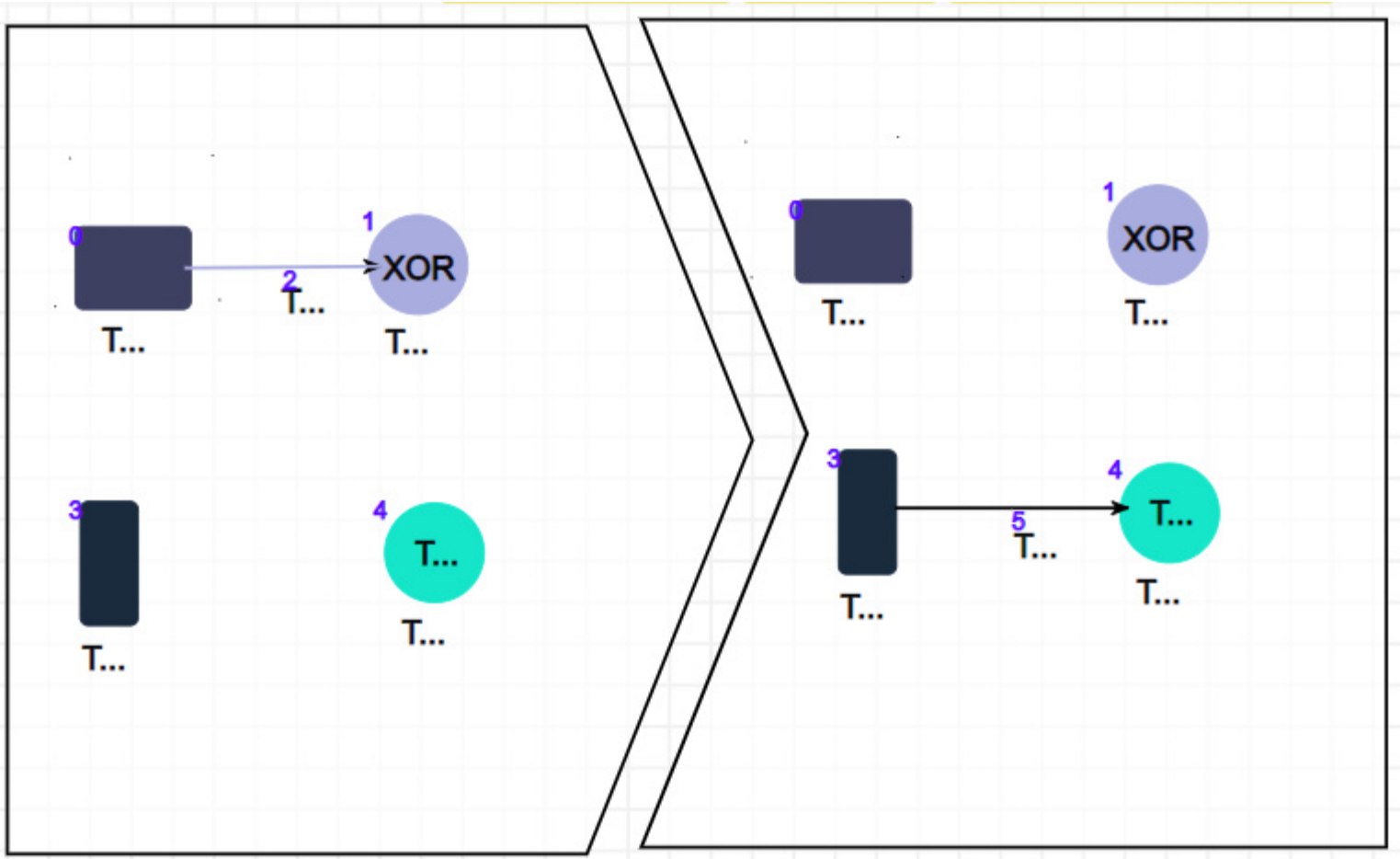
- **Règle 3 R_xor2place:**

Génère une nouvelle place pour chaque connecteur XOR.



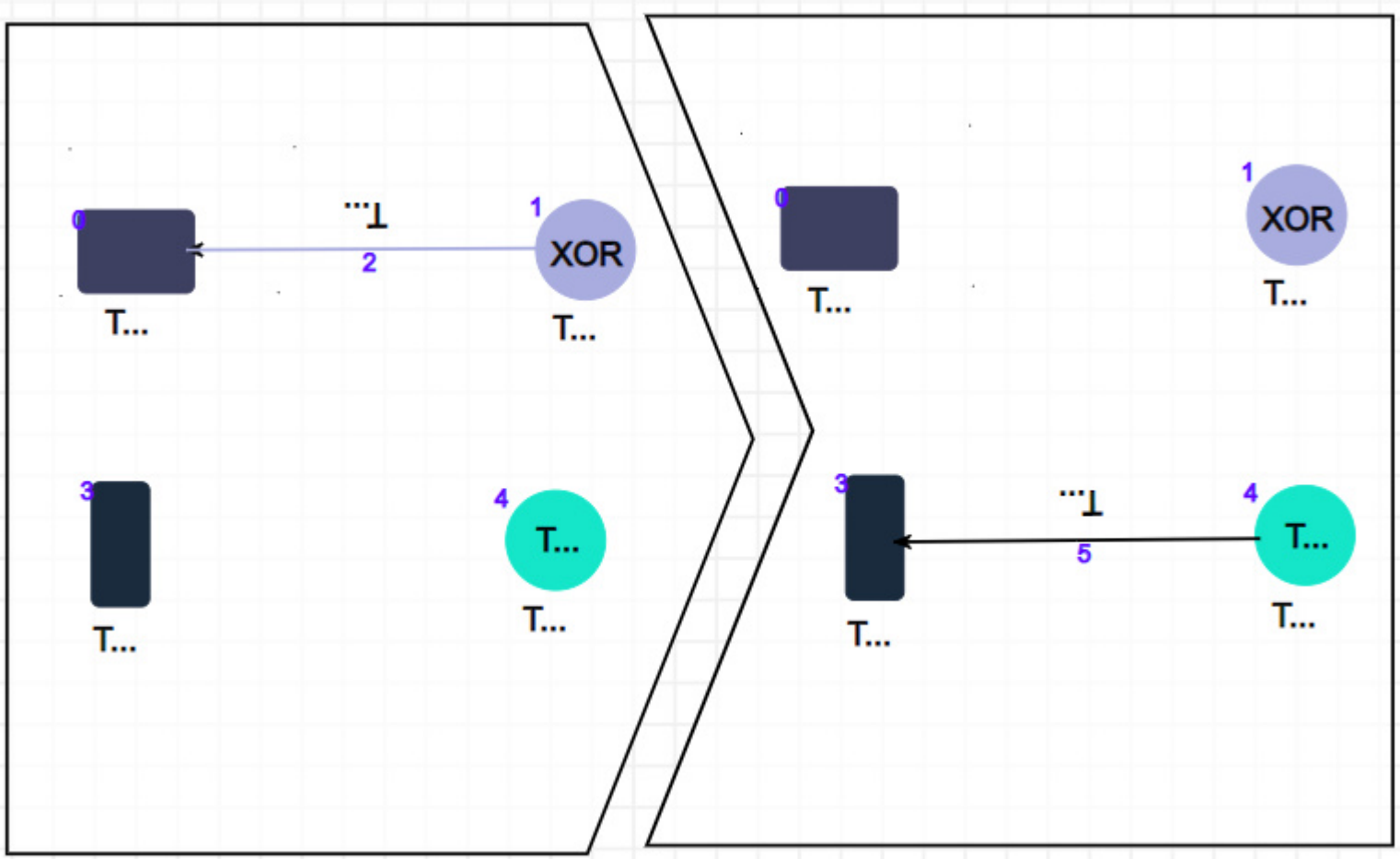
Règles de la transformation BPMN vers RDP

- **Règle 4 R_inXOR2inPL:**
  - Génère un arc sortant de la transition (attachée à l'activité) vers la place (attachée au lien XOR)
  - Supprime tous les arcs (Act2XOR)



Règles de la transformation BPMN vers RDP

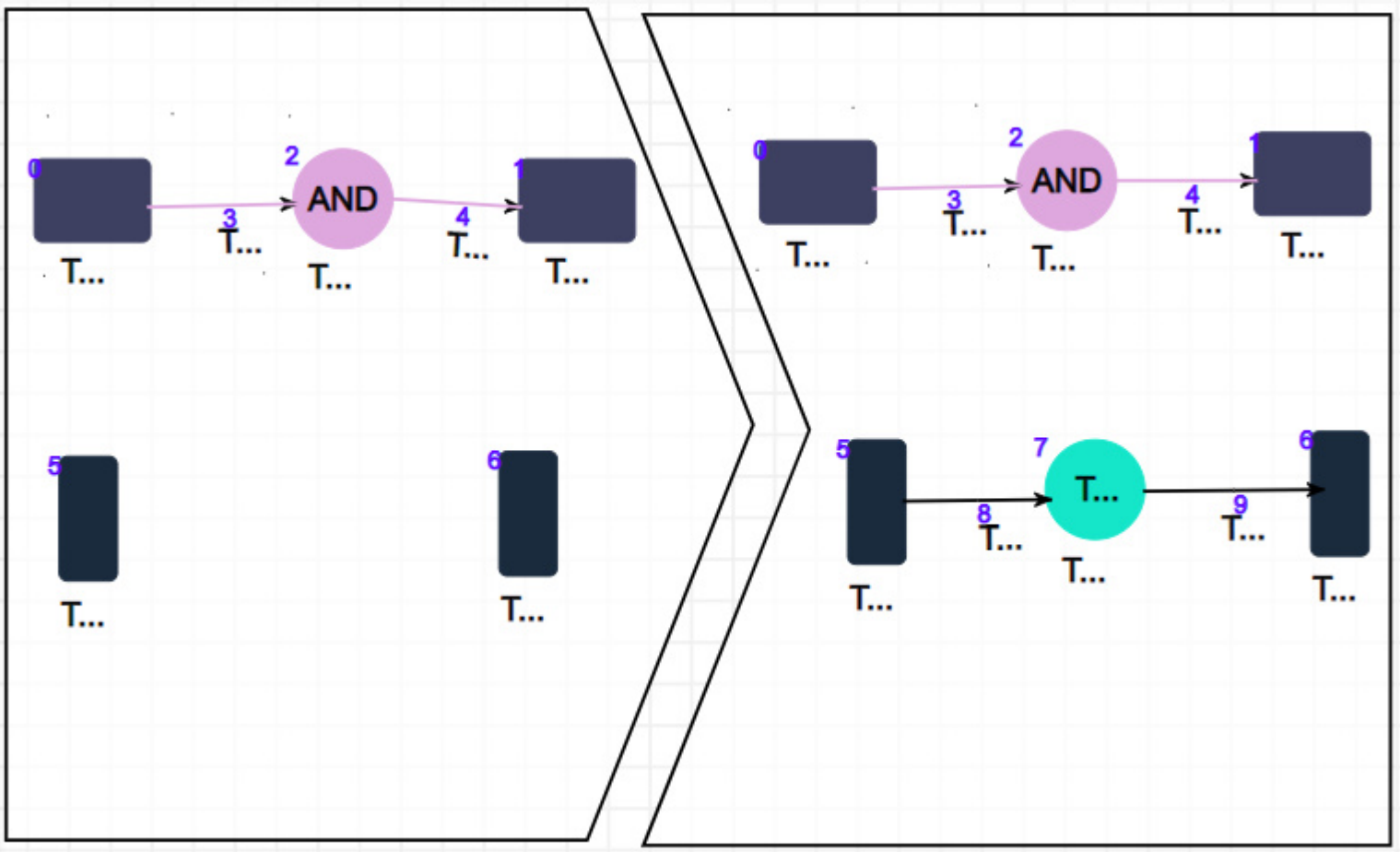
- **Règle 5 R_outXOR2outPL:**
  - Génère un arc sortant de la place (attachée au lien XOR) vers la transition (attachée à l'activité)
  - Supprime tous les arcs (XOR2Act)



Règles de la transformation BPMN vers RDP

- **Règle 6 R_ANDrule:**

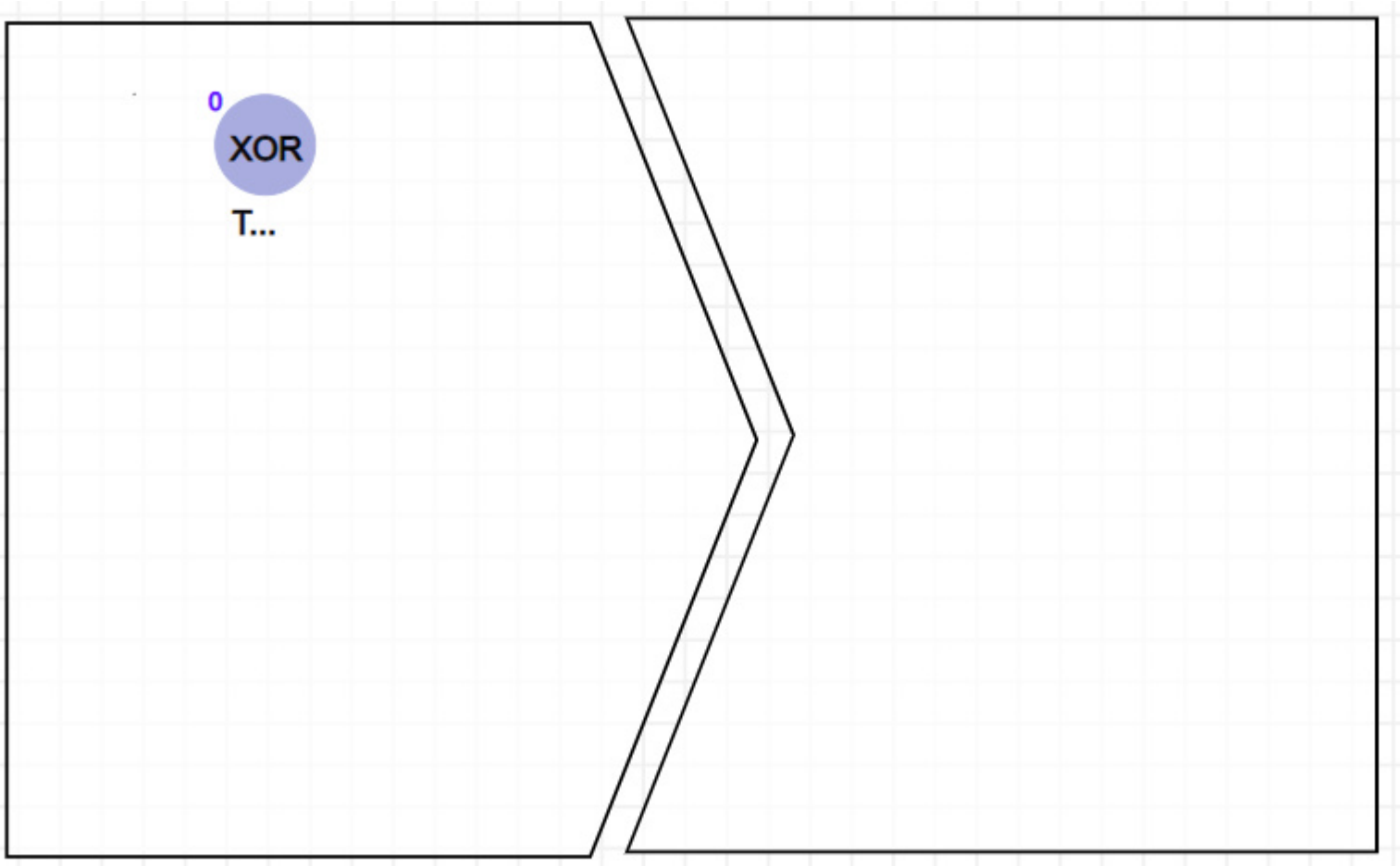
Génère une nouvelle place avec un arc entrant et un arc sortant aux transitions (Liés aux activités)



Règles de la transformation BPMN vers RDP

- **Règle 7 R_DeleteXOR:**

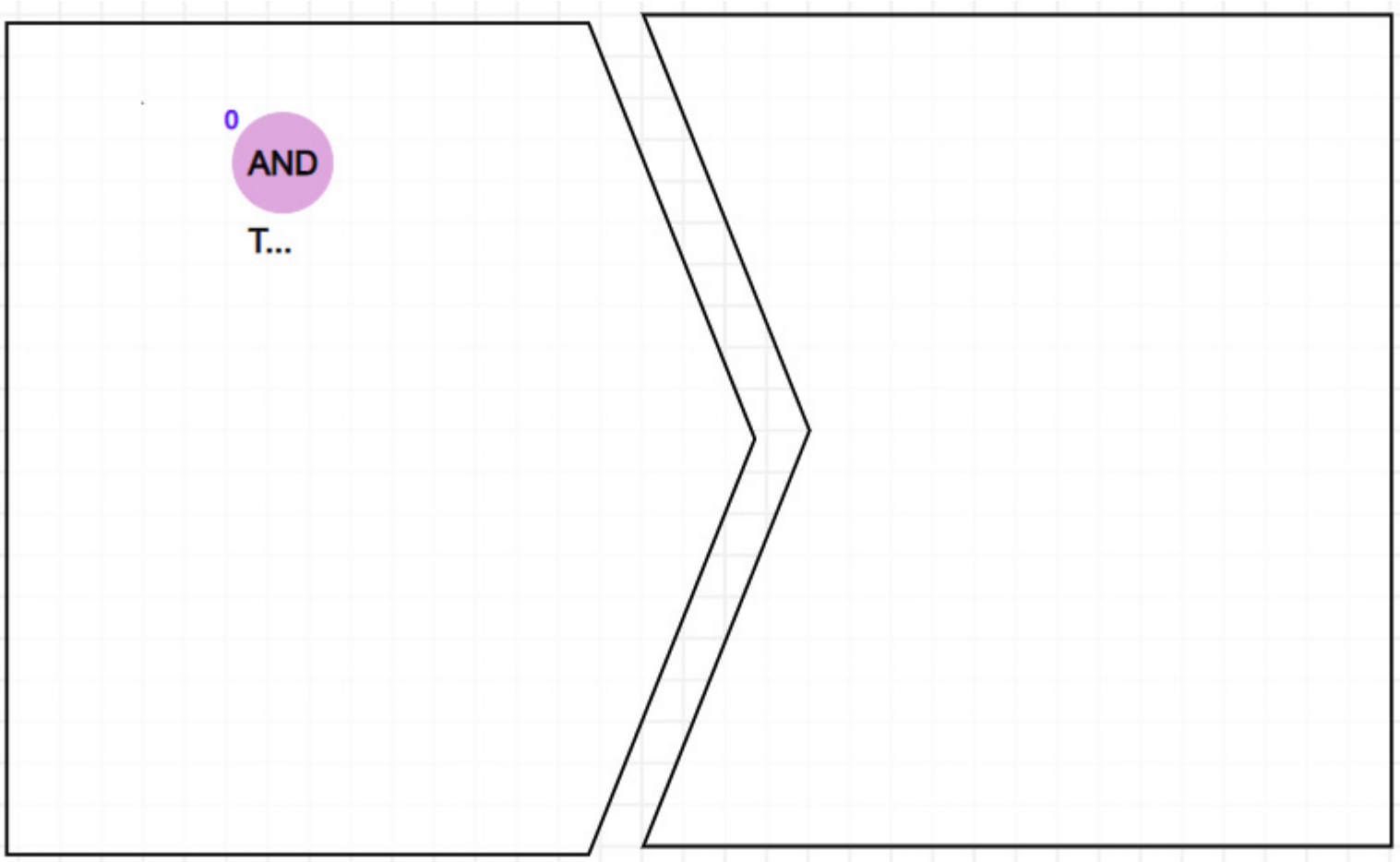
Supprime tous les connecteurs XOR du modèle BPMN



Règles de la transformation BPMN vers RDP

Règle 8 R_DeleteAND:

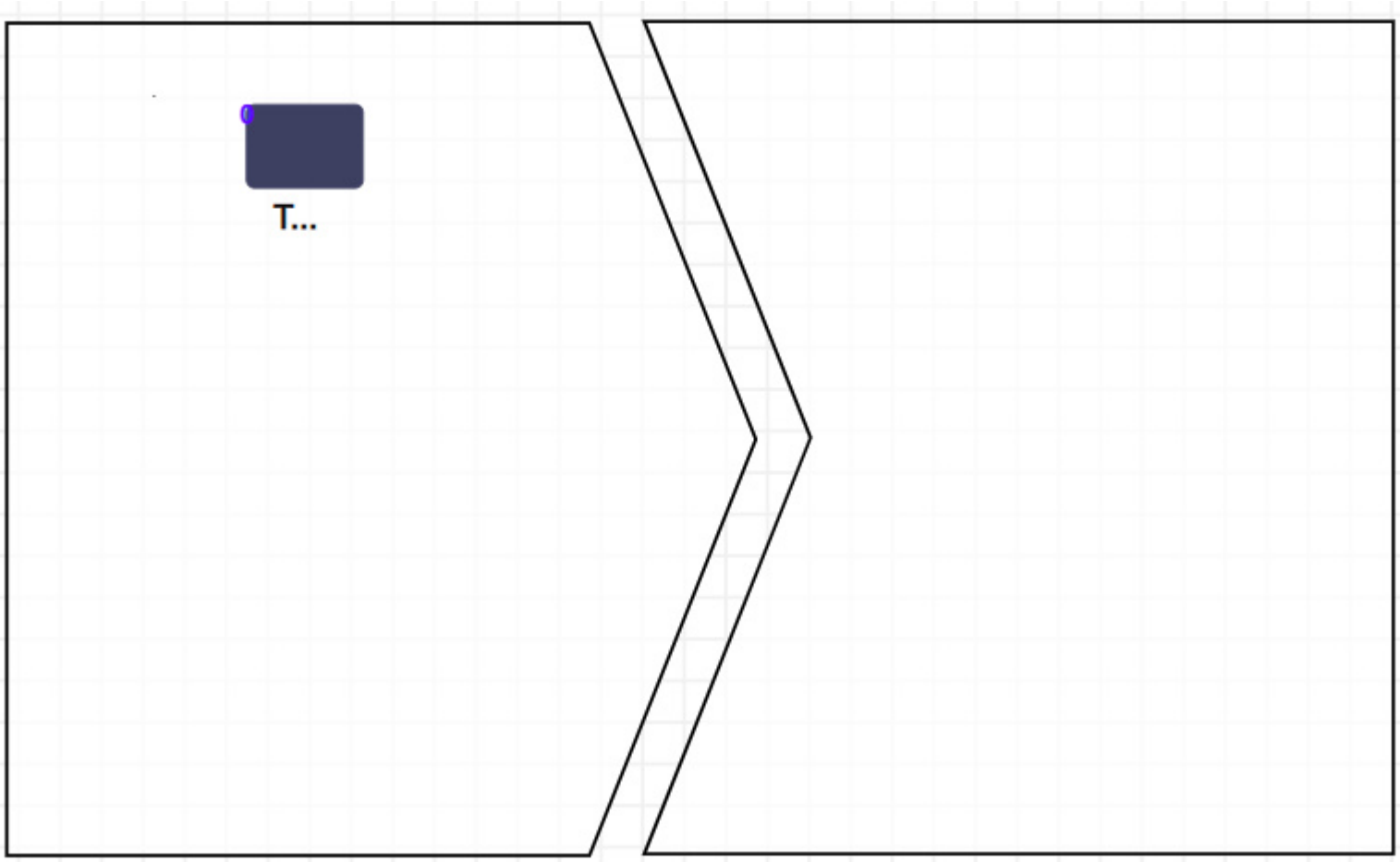
Supprime tous les connecteurs AND du modèle BPMN.



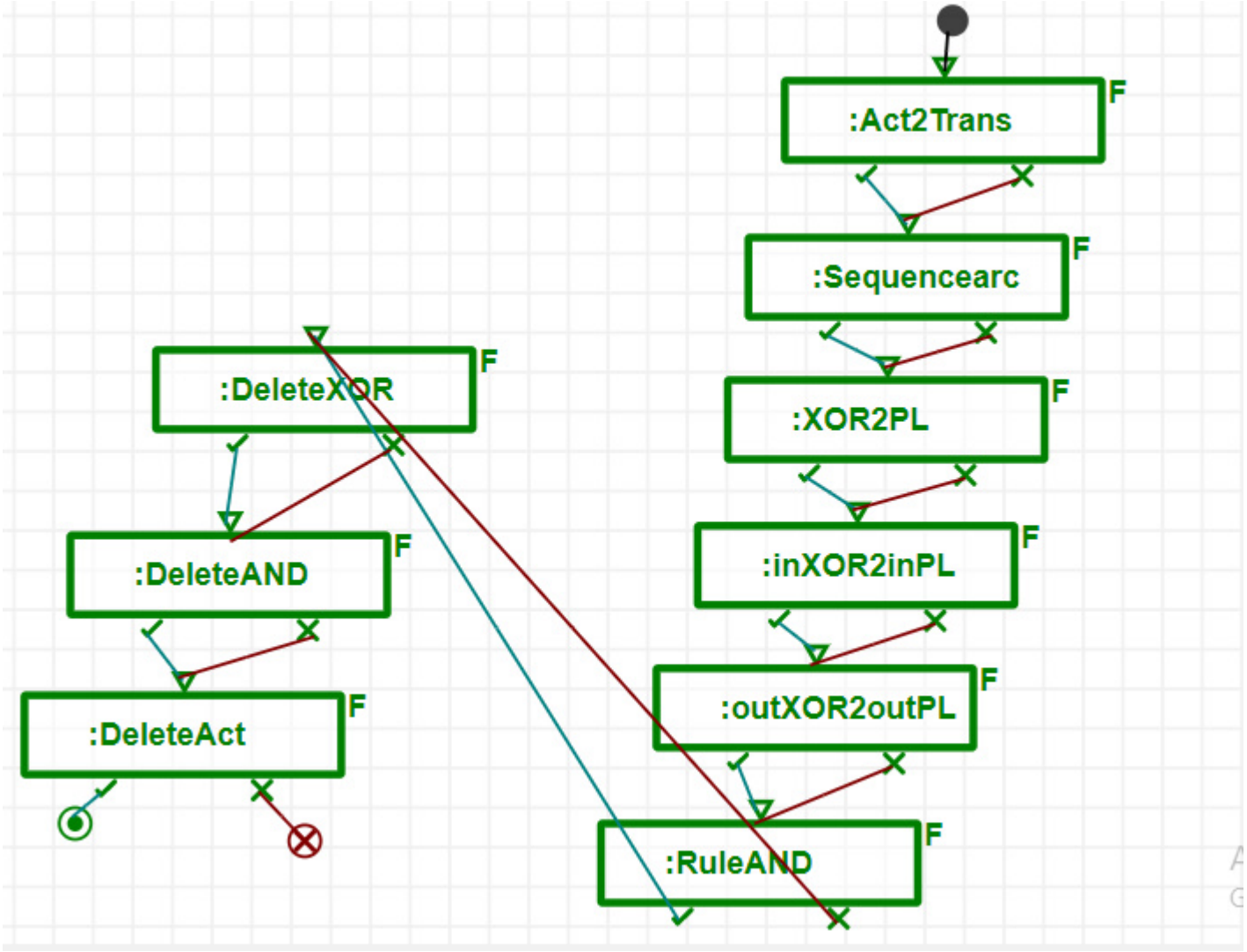
Règles de la transformation BPMN vers RDP

Règle 9 R_DeleteActivity:

Supprime tous les activités du modèle BPMN



Processus d'exécution des règles dans l'outil AToMPM

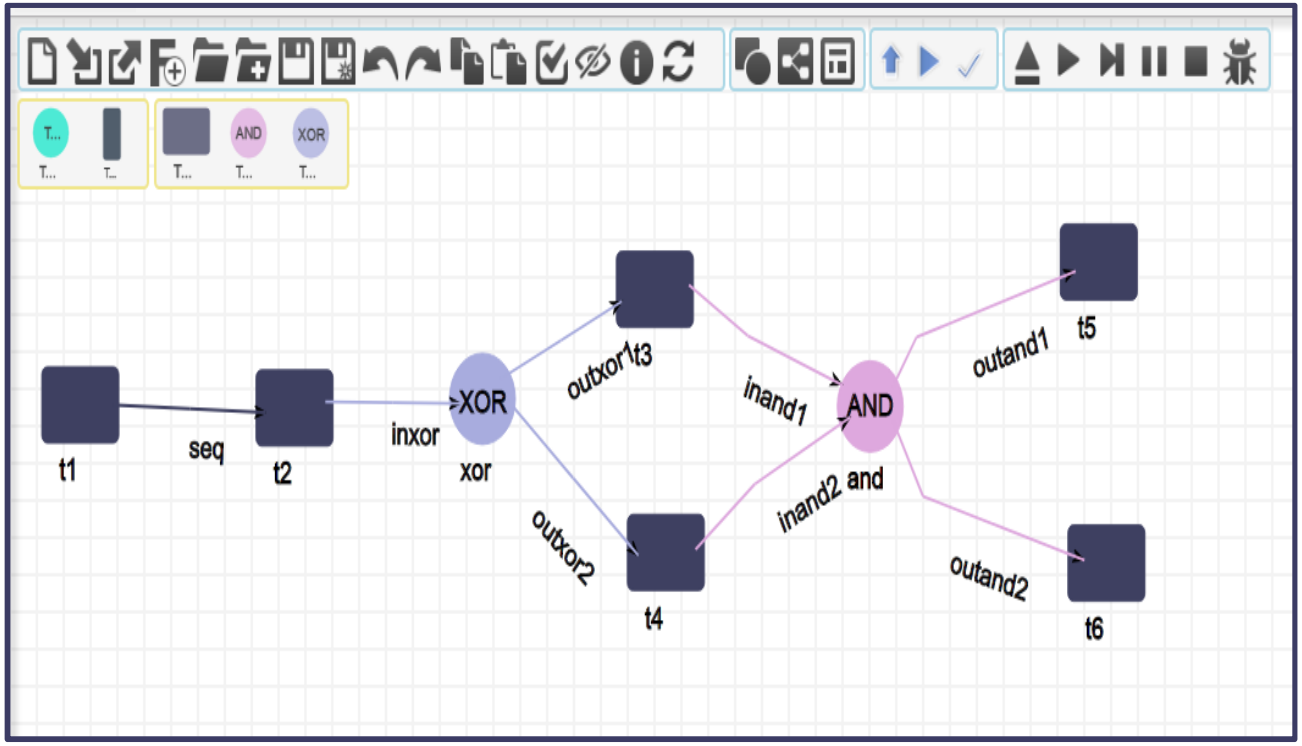


Plan du chapitre

- Concepts de Transformation de Graphe
  - Principes de L'IDM
  - Transformation de modèle
  - Transformation de graphe
  - L'outil AToMPM
- Transformation des BPMN vers RdP
  - Les formalismes source BPMN et cible Rdp
  - Méta-modélisation
  - Règles de transformation
- **Exemple de Transformation d'un modèle BPMN**
- Conclusion

Exemple de Transformation d'un modèle BPMN

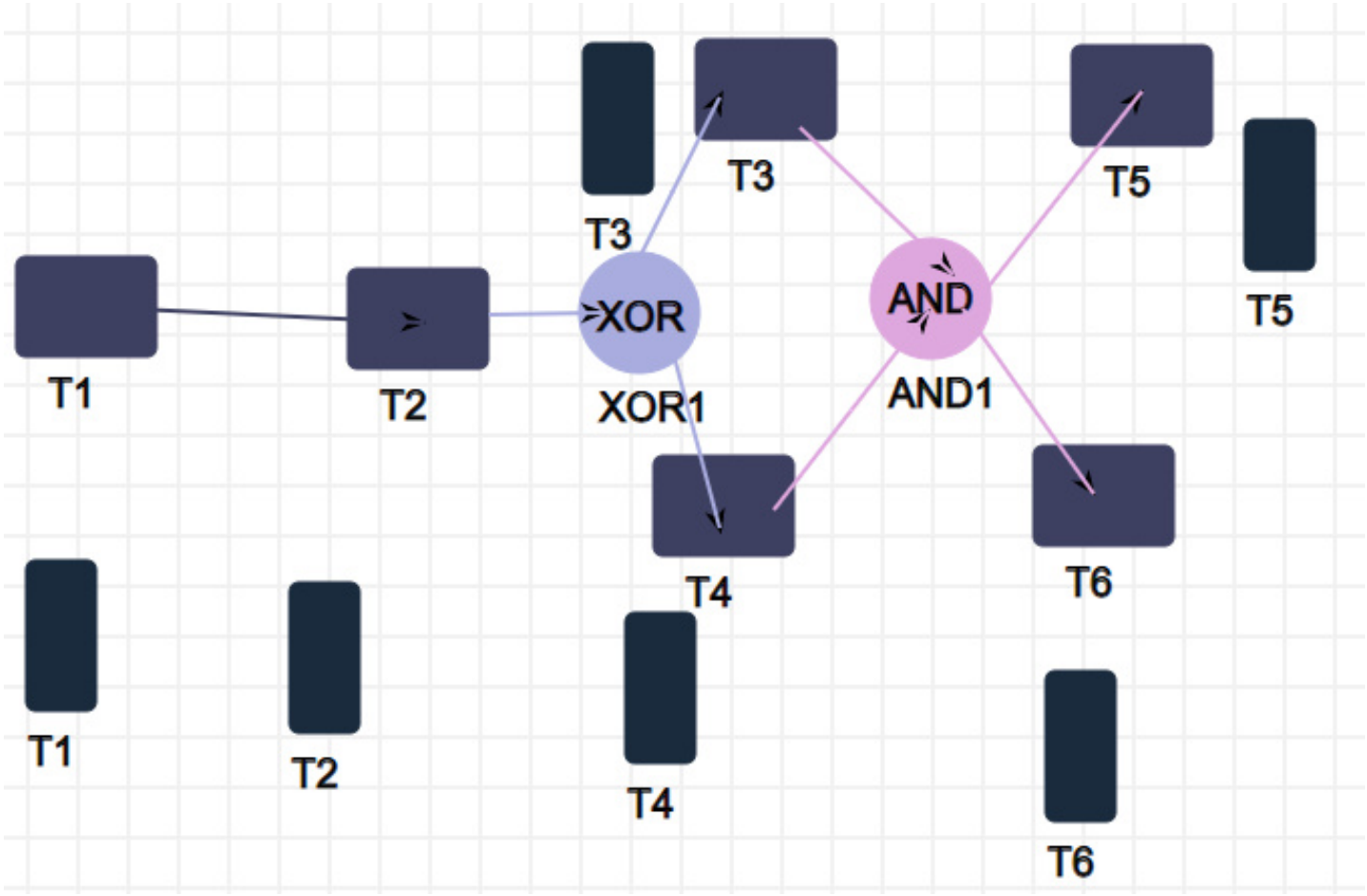
Modèle BPMN source



Exemple de Transformation d'un modèle BPMN

Modèle Intermédiaire après application:

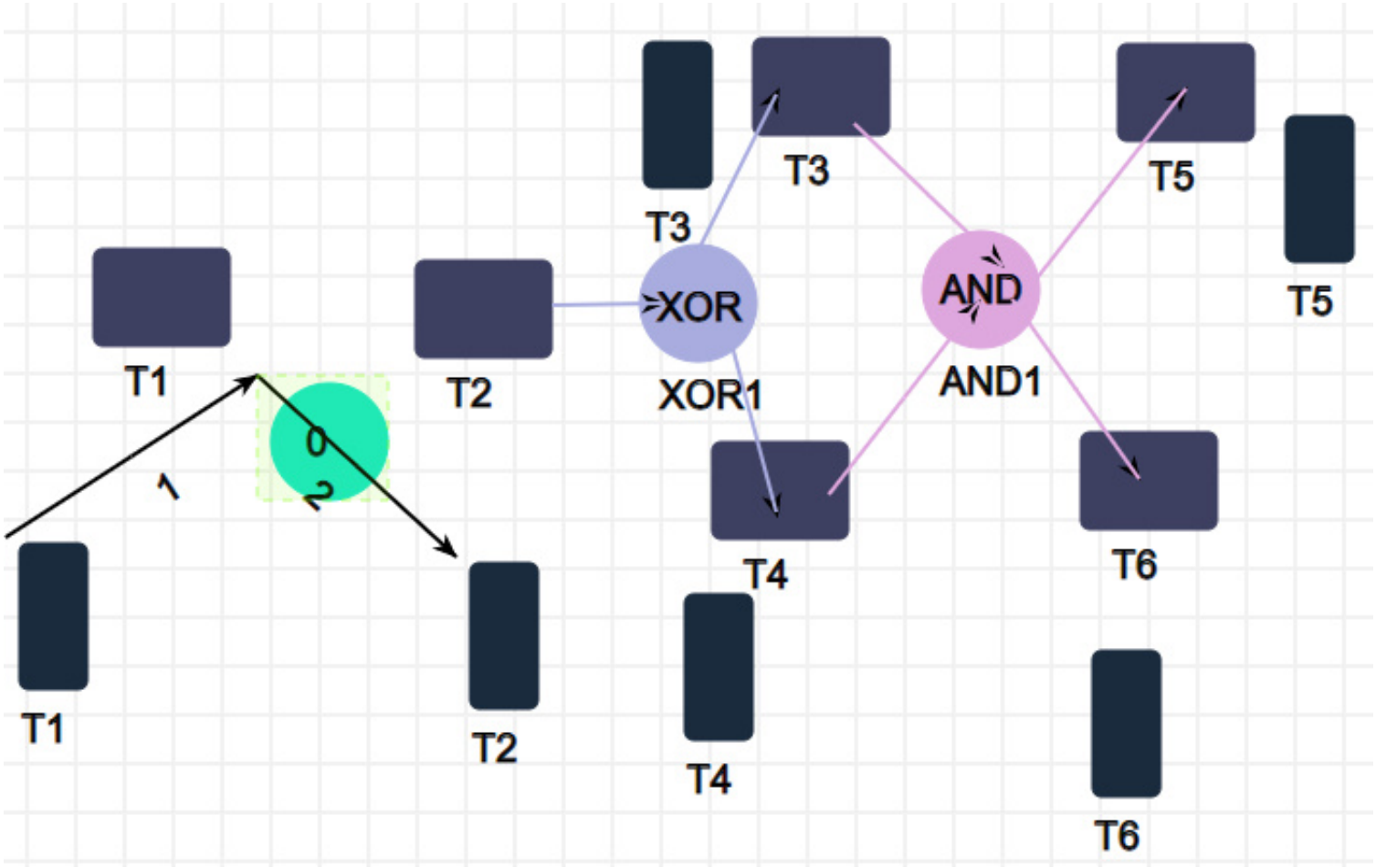
Règle 1_R_Activity2Transition



Exemple de Transformation d'un modèle BPMN

Modèle Intermédiaire après application:

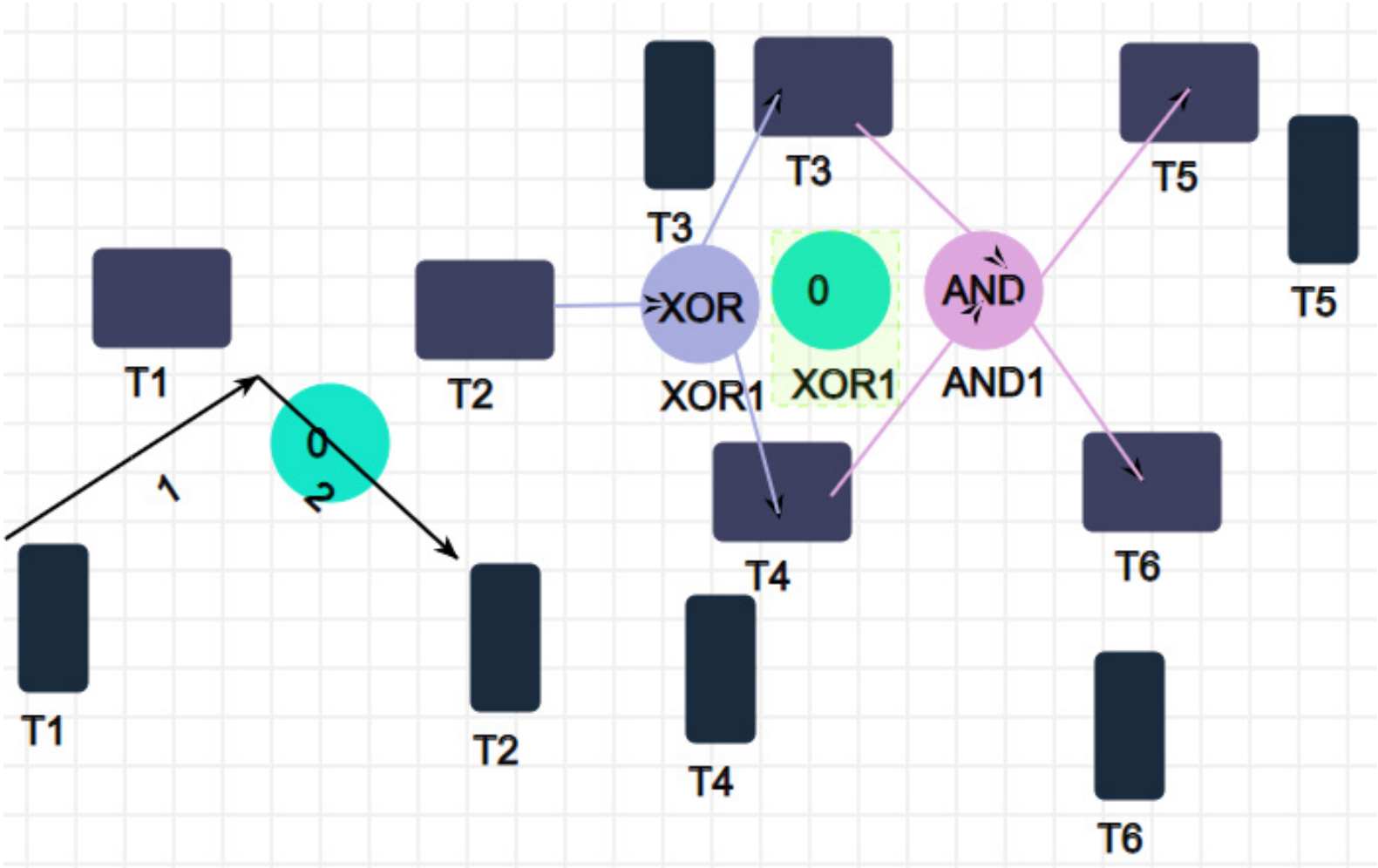
Règle 2 R_sequencearc



Exemple de Transformation d'un modèle BPMN

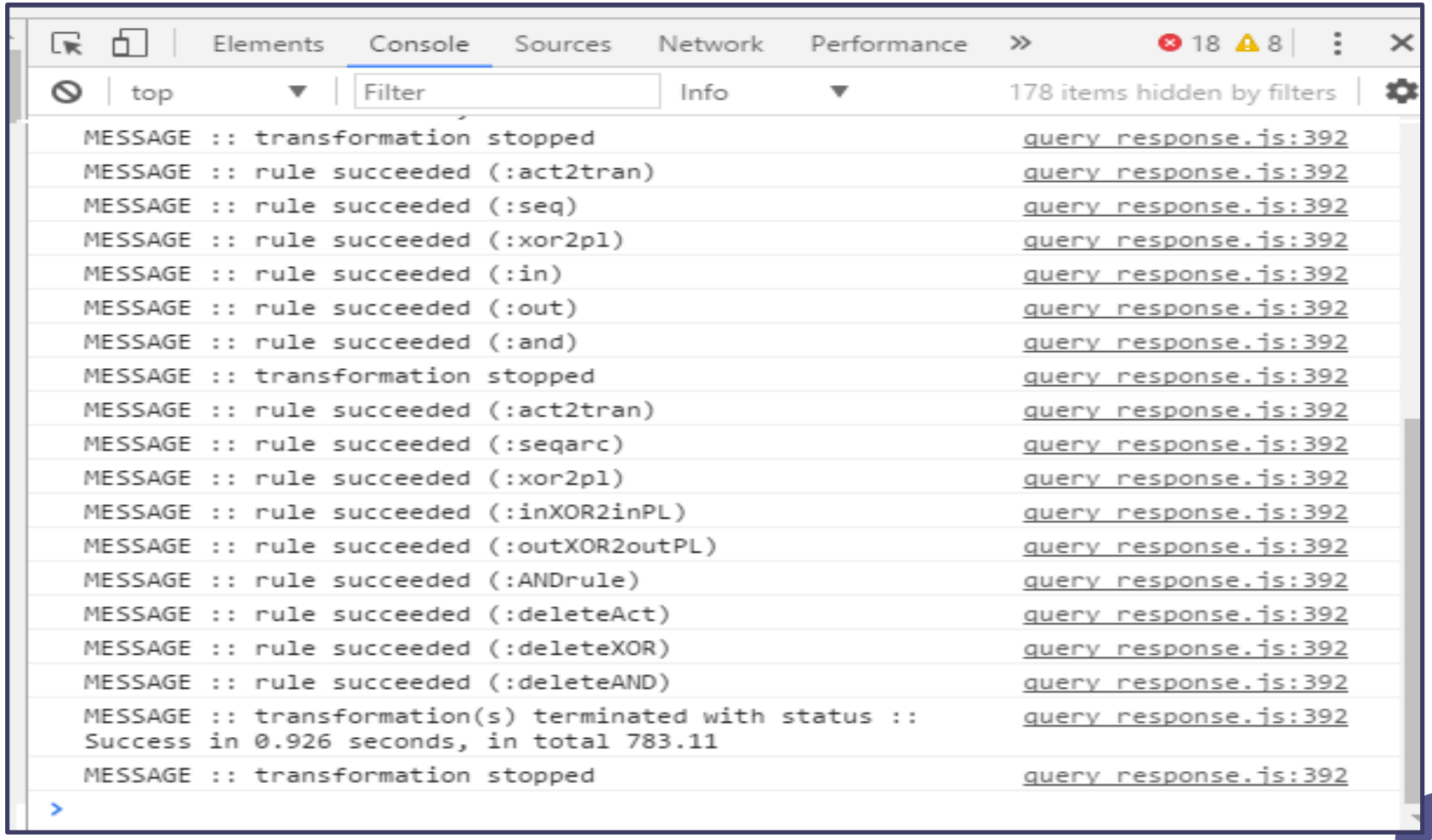
Modèle Intermédiaire après application:

Règle 3 : R_xor2place



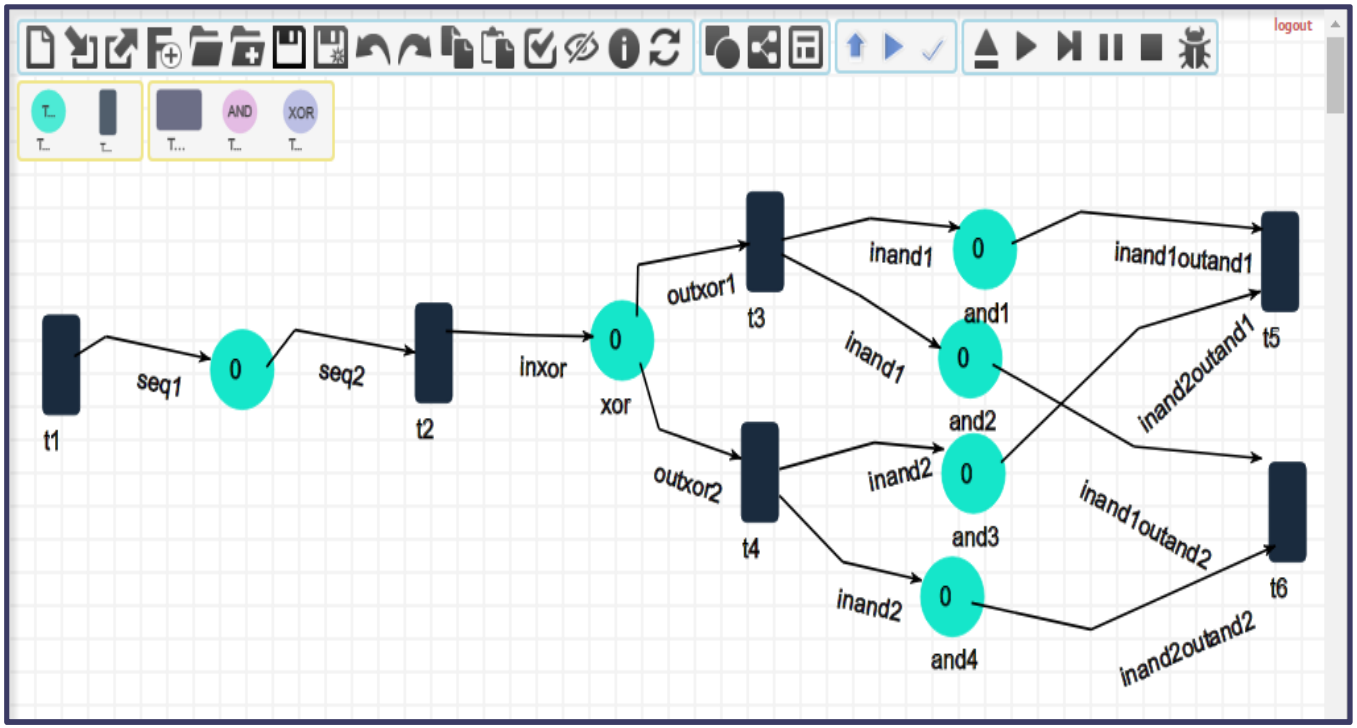
Exemple de Transformation d'un modèle BPMN

Exécution de la transformation



Exemple de Transformation d'un modèle BPMN

Modèle RdP cible



Conclusion

Dans ce chapitre, nous avons présenté les principes des transformation de graphes en utilisant une étude de cas.

-Premièrement, nous avons présenté les concepts théorique :

- ✓ IDM, Méta-modélisation et transformation de modèles, notion de graphe , transformation de graphes, réécriture de graphe, grammaire de graphe…

-Deuxièmement, nous avons proposé une Transformation des BPMN vers les RdP à l'aide de l'outil AToMPM.

✓ Le travail est réalisé en trois étapes :

- ❖ Définir un méta-modèles du langage BPMN, et un méta-modèles des réseaux de Pétri
- ❖ Définir une grammaire de graphe pour effectuer la transformation d'un modèle BPMN vers leur équivalent en RDP.
- ❖ Appliquer l'approche proposée sur un exemple

Références

1. ***Thèse de Doctorat en sciences de Mr Kerkouche Elhillali (Université Mentouri Constantine, 2010)***

2. ***Cours de Pr. Alloua Chaoui, Université de Constantine 2***

3. ***Memoire de Master RSD en Informatique de Makhloufi & Benkara (Université de Constantine 2, 2017)***

4. ***AToMPM home page : https://atompm.github.io/***