Méthodes avancées de programmation et de développement logiciel

Conception finale (séance TD3)

Auteurs :
M. Yann LERQUEMAIN, étudiant FISE A2 IMT ATLANTIQUE
yann.lerquemain@imt-atlantique.net
M. Quentin RABILLOUD, étudiant FISE A2 IMT ATLANTIQUE
quentin.rabilloud@imt-atlantique.net

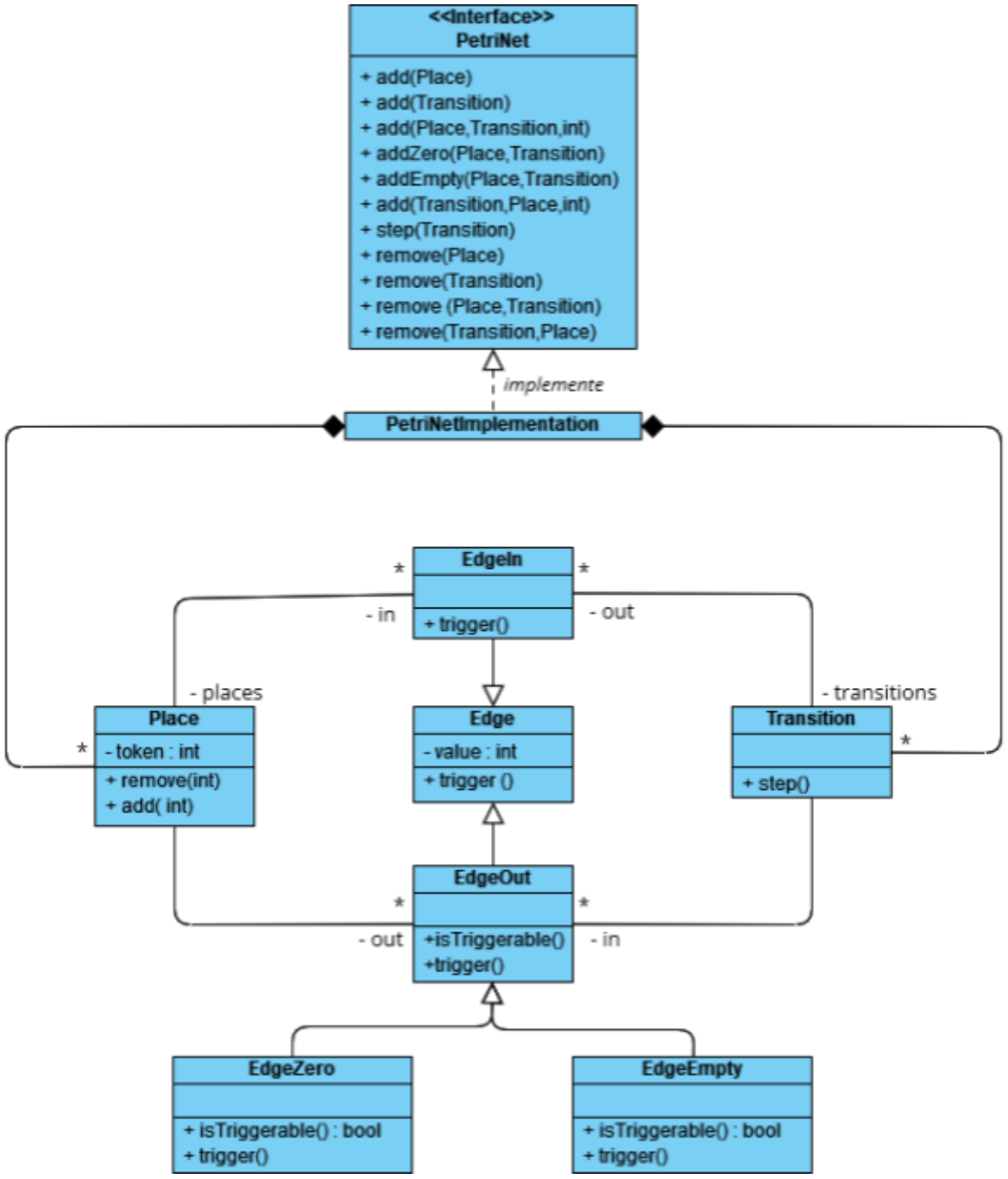
Destinataires :
M. Matthew COYLE, doctorant IMT ATLANTIQUE
matthew.coyle@imt-atlantique.fr

Le 04 octobre 2023

version 1.0

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Diagramme de classe détaillé retenu



Justification des choix du diagramme de classe

Interface PetriNet :

Il semble pertinent de définir une interface qui fixe les exigences pour la construction d'une classe représentant un réseau de petri. Cette interface permet de garantir que l'implémentation choisie contient toutes les méthodes nécessaires à la création, l'initialisation et la modification d'un réseau de petri par un usager.

Classe PetriNetImplementation :

À partir de l'interface définie ci-dessus, on construit une classe permettant d'implémenter le réseau de petri dans sa globalité. Cette classe sert de conteneur principal pour les différents éléments composant un réseau de petri.

Classe Transition :

Dans ce choix d'implémentation, la transition est au centre du fonctionnement du PetriNet, chaque transition est reliée à un nombre arbitraire d'arcs en sortie de places et à un nombre arbitraire d'arcs en entrée de places. Il peut y avoir autant de transitions qu'on le souhaite dans le réseau de petri.

Classe Edge :

La classe Edge est une classe mère permettant de définir les attributs et les méthodes partagées par tous les arcs, bien que certains soient entrants et d'autres entrants. Chaque arc est caractérisé par un nombre de jetons.

Classe EdgeOut :

Cette classe est directement héritée de Edge, elle représente les arcs en sortie de places. Dès lors, elle a une méthode supplémentaire qui permet de vérifier si les places en amont contiennent suffisamment de jetons ou non pour déclencher la transition. Chaque EdgeOut est reliée à une et unique transition, et à une et unique place.

Classes EdgeZero et EdgeEmpty :

Ces deux classe permettent d'ajouter deux nouveaux types d'arc au réseau de petri. Comme ces deux types d'arc se trouvent en sortie de place, ils héritent de la classe EdgeOut, ce qui permet de les ranger par généricité dans un même objet conteneur que les autres arcs sortants. Les arcs de type EdgeZero s'activent uniquement lorsque la place en amont est vide, tandis que les arcs de type EdgeEmpty s'activent lorsqu'il y a au moins un jeton dans la place en amont et provoquent le retrait de tous les jetons présents.

Classe Place :

Les places représentent les lieux de stockage des jetons, elles contiennent toutes un nombre défini de jeton, qui varie au fur et à mesure des évolutions du réseau. Il est possible de retirer des jetons à une place, par exemple si elle se trouve en amont et que les arcs sortants sont activés, ou d'ajouter des jetons à une place en aval, si une transition est tirée. Chaque place est reliée à un nombre arbitraire d'arcs sortants et d'arcs entrants.

Classe EdgeIn :

Cette classe hérite également de Edge, cependant elle permet de spécifier le comportement des arcs entrants dans une place. Chaque arc est relié à une unique transition et à une unique place. Les arcs entrants sont activés si la transition à laquelle ils sont reliés est tirée.

Diagramme de séquence 1 : Trigger d'un step sur une transition d'une unique place à une unique autre

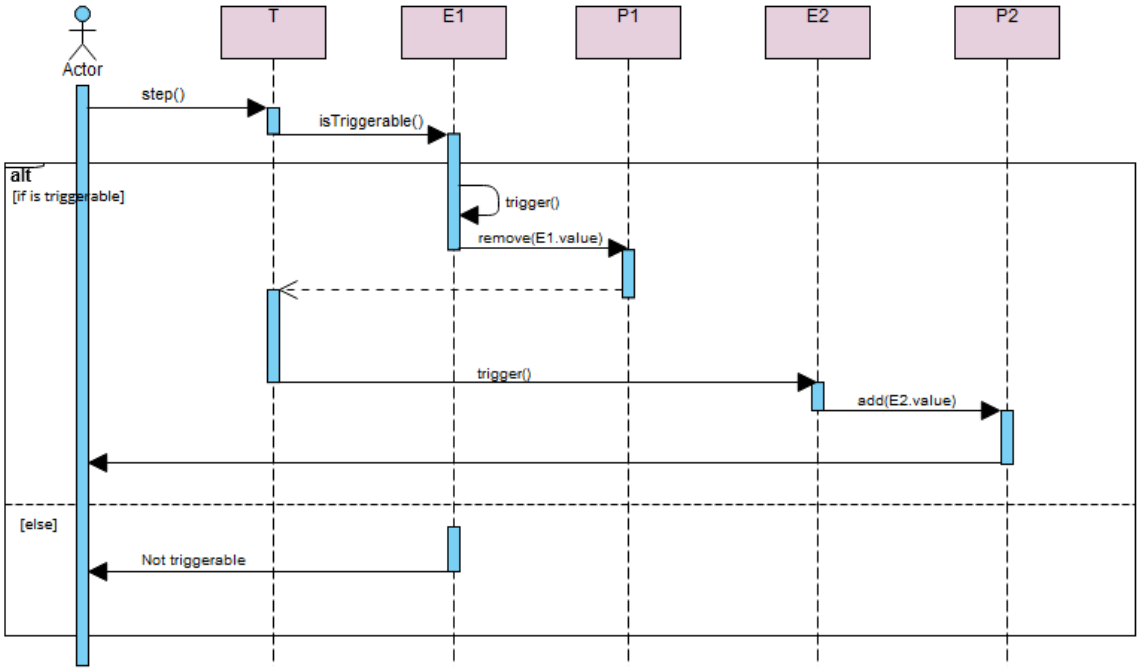
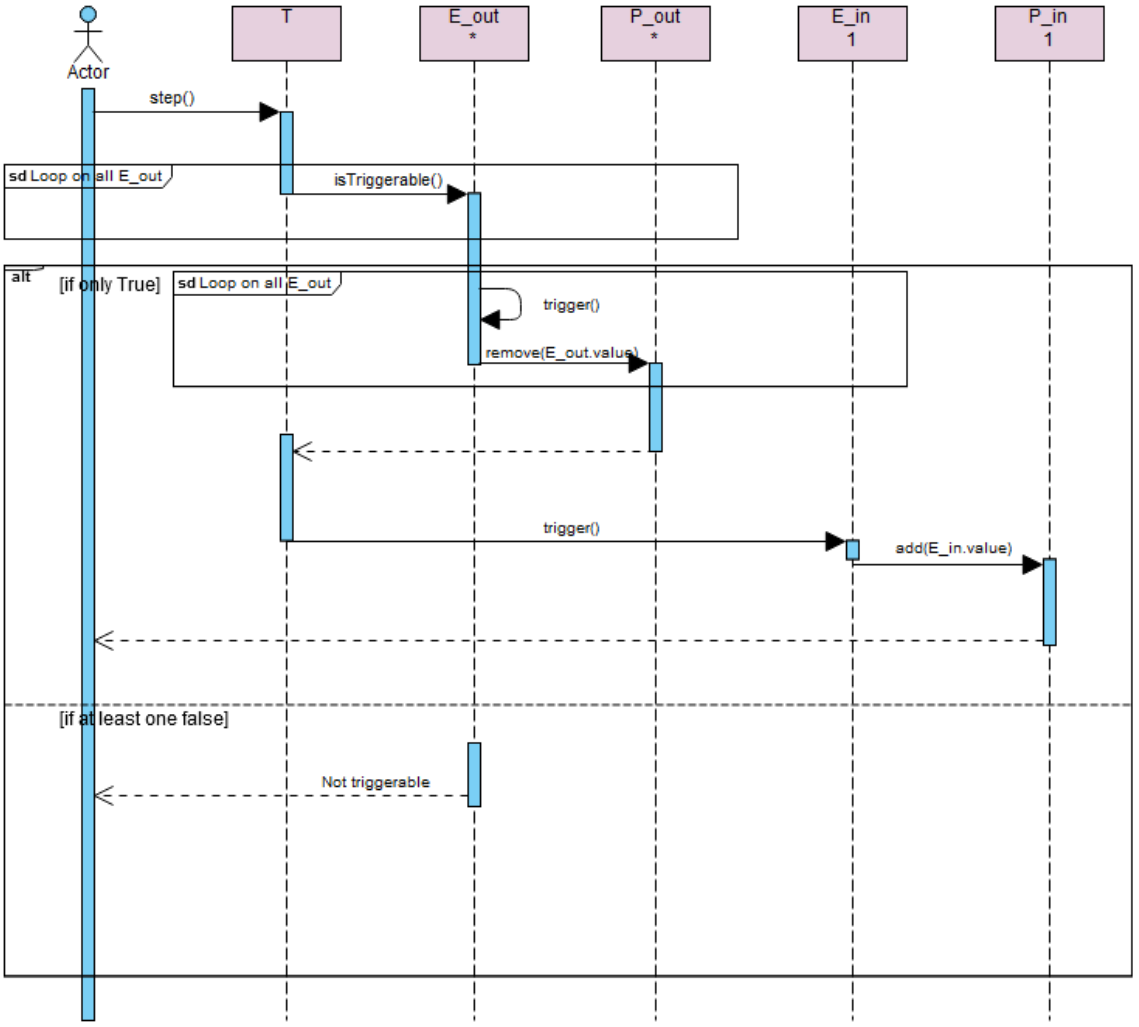


Diagramme de séquence 2 : Généralisation du diagramme de séquence 1 à plusieurs places d'entrée, et une unique place de sortie



Annexe modifications :

TP3-4 11/10

Dans le diagramme de classe :

Interface :

- remove(place,transition) devient remove(EdgeOut)
- remove(transition,place) devient remove(EdgeIn)