



Génie Logiciel et projet de synthèse

Analyse dynamique

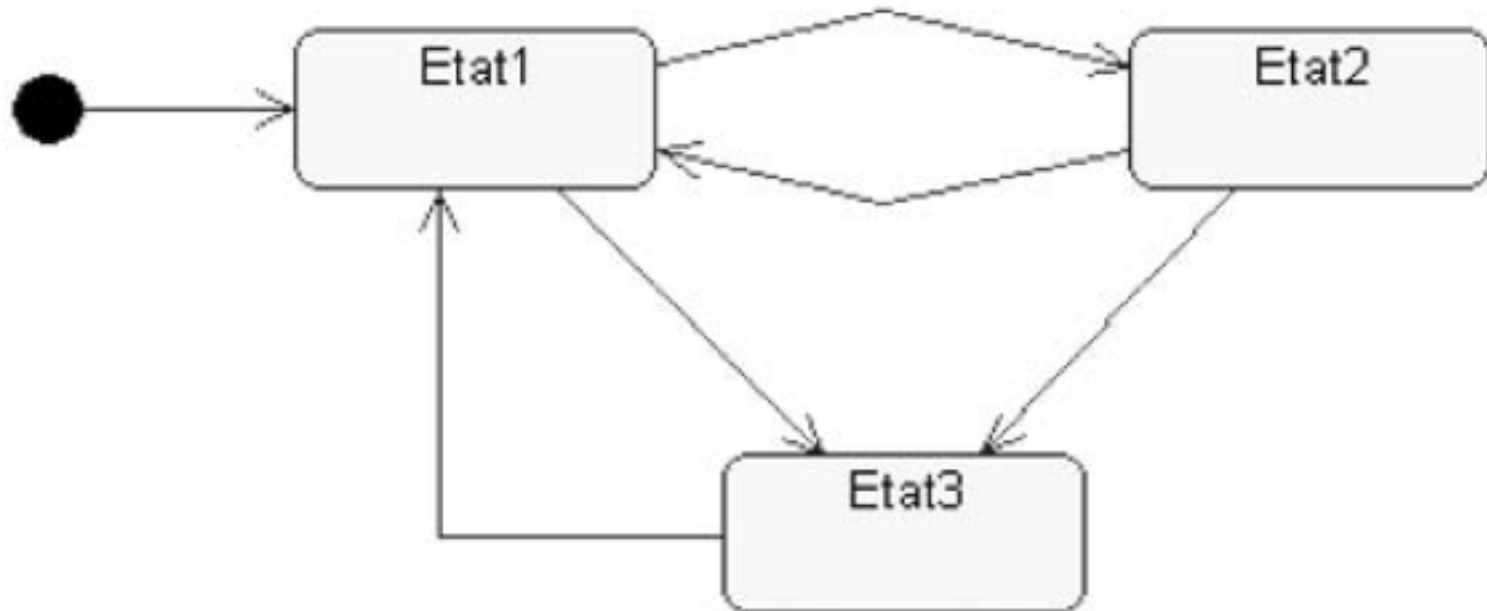
Kévin Bailly
Institut des Systèmes Intelligents et de
Robotique

kevin.bailly@upmc.fr

- **Diagramme d'états**

- Concept de la **machine à états finis** : analyse du **cycle de vie** d'une instance de classe
- Pour les classes avec un comportement dynamique complexe
- Etape de construction du diagramme
 - 1 - représenter les **états** et les **transitions** du comportement nominal
 - 2 – ajouter les comportements alternatifs
 - 3 – ajouter les **effets** (sur les transition) et les **activités** (dans les états)
 - 4 – subdiviser en sous états si trop complexe

- **Diagramme d'états : représentation**



- **Etat**

- Etat = **situation** d'un objet à un moment de son cycle de vie
- L'état de l'objet est fonction des **événements**

- **Comment identifier les états d'une classe ?**

- Pas facile !
 - **Recherche intuitive** : dépend de la connaissance du métier → parler avec l'expert
 - Analyse des **attributs** et des **associations** d'une classe : qu'est ce qui modifie son état ?
 - Pour tous les scénarios de la classe : analyse les **interactions** des instances de la classe et identifier les états entre chaque interaction

- **Transition :**
 - Réaction de l'objet à un événement
 - Généralement : changement d'état
- **Une transition possède :**
 - Un événement déclencheur
 - Une condition de garde
 - Un effet
 - Un état cible

- **Evènement**

- occurrence d'un **stimulus** qui peut déclencher une **transition**
- En UML, 4 sortes d'évènements
 - Réception d'un **signal** envoyé par un autre objet (asynchrone)
 - **Appel** d'une opération sur l'objet (synchrone)
 - **Changement** dans la satisfaction d'une condition :
when
 - **Ecoulement du temps** : after
- Un événement peut porter **des paramètres** :
transmission d'informations entre objets

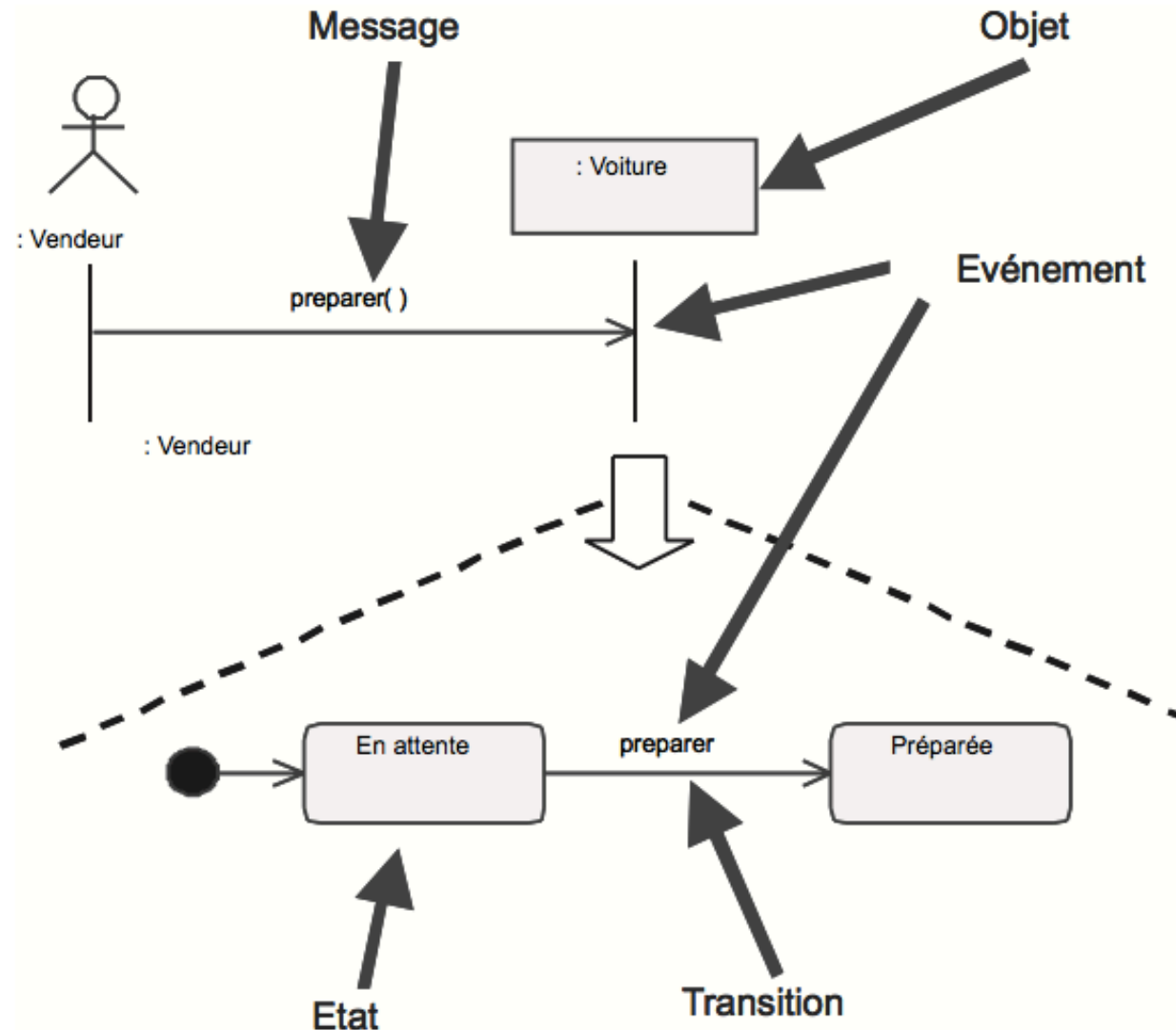
- **Message**

- Transmission d'informations **unidirectionnelle** entre deux objets
- Mode synchrone ou asynchrone
- Réception d'un message = événement qui doit être traité par le récepteur

- **Condition (de garde)**

- Expression booléenne
- Doit être vraie pour que la transition soit déclenchée
- Peut dépendre
 - d'attributs de l'objet
 - de paramètres de l'événement déclencheur

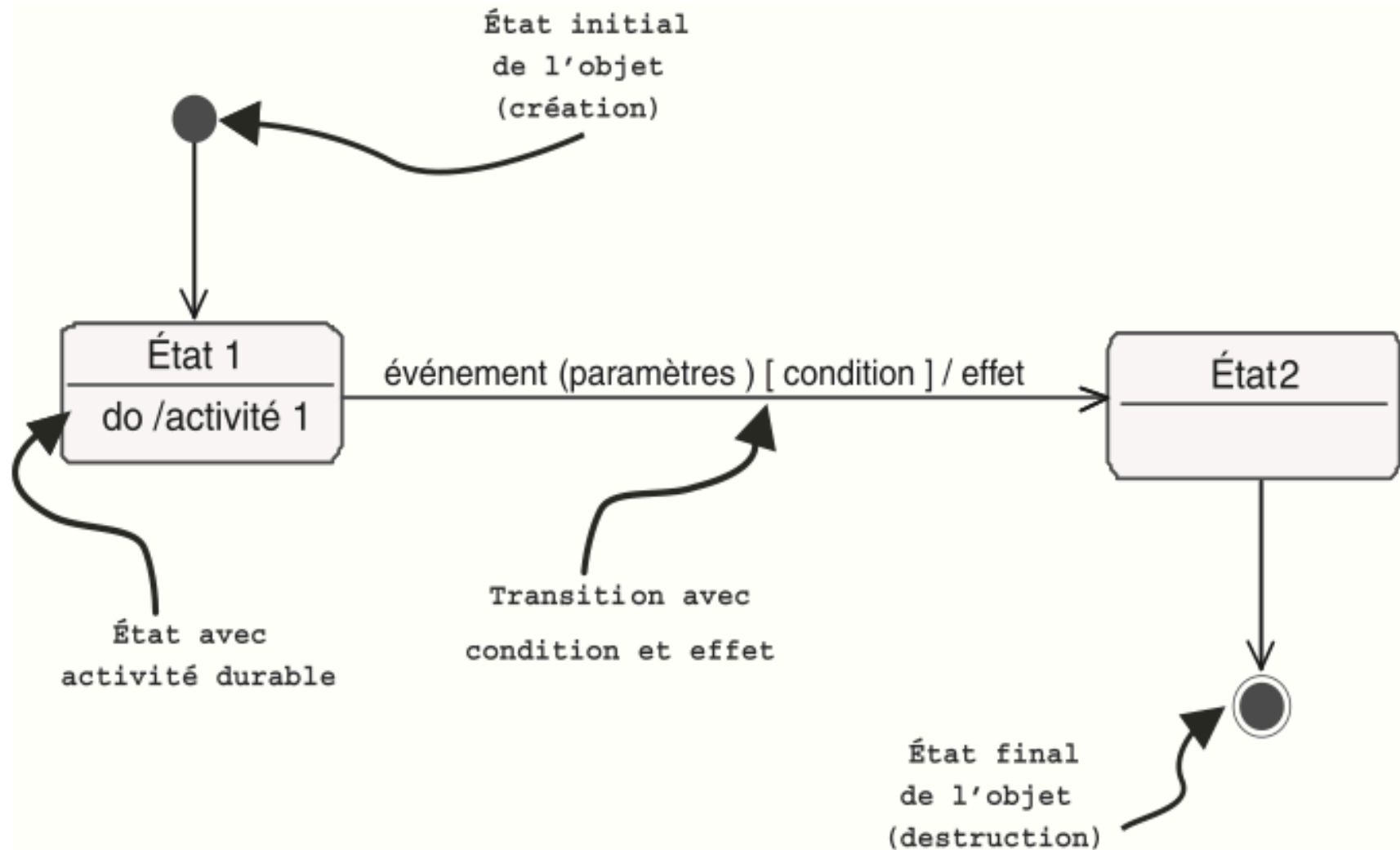
Représentation des concepts dynamiques



- **Effet**

- Comportement optionnel d'une transition
- Par exemple : une action ou une séquence d'actions :
 - Mise à jour d'un attribut
 - Création d'un objet
 - Envoi d'un signal à un autre objet
 - ...

Représentation des diagrammes d'états

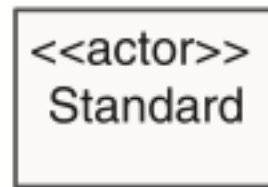


Etude d'un téléphone public

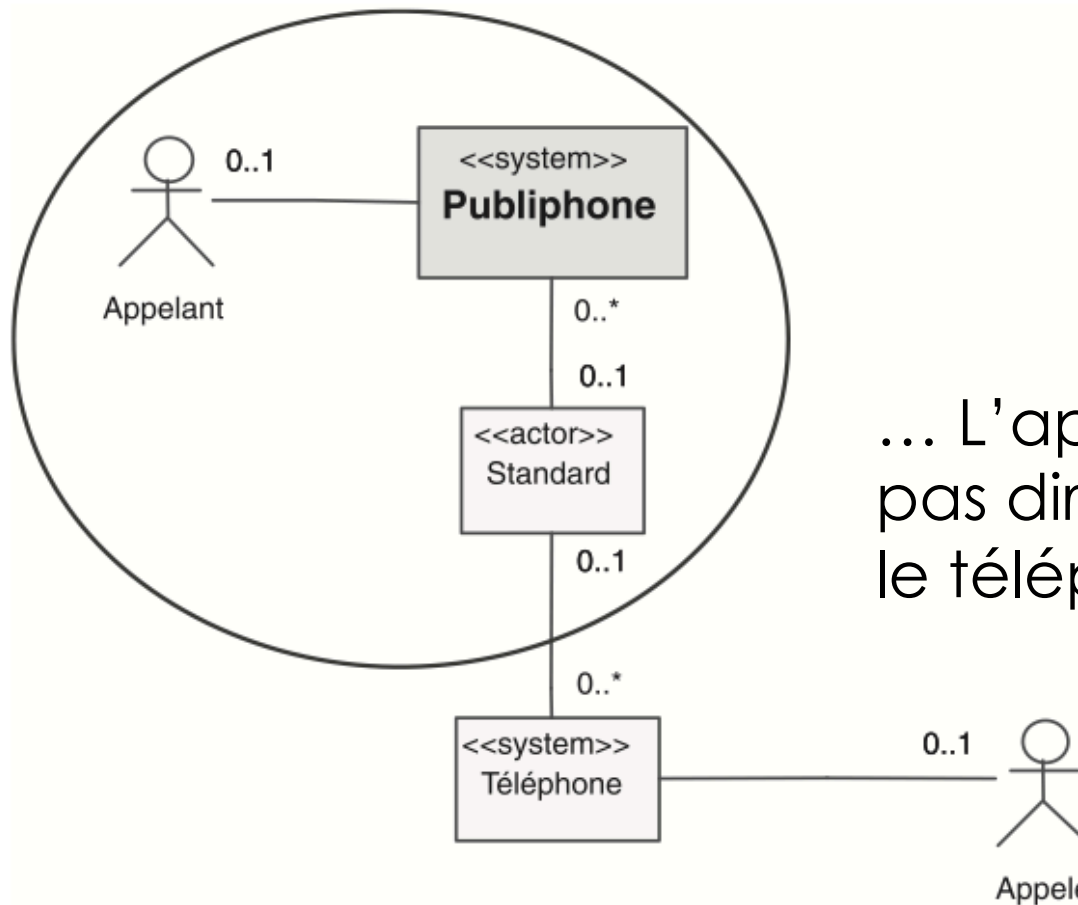
- 1 le prix minimal d'une communication nationale est de 20 ct.
- 2 Après introduction de la monnaie, l'utilisateur à 2 minutes pour composer son numéro (délai décompté par le standard)
- 3 La ligne peut être libre ou occupée
- 4 Le correspondant peut raccrocher le premier
- 5 Le téléphone consomme de l'argent dès que l'appelé décroche et à chaque unité de temps (géré par le standard)
- 6 On peut ajouter des pièces à tout moment
- 7 Lors du raccrochage, le solde de monnaie est rendu

Identifier les acteurs

- Acteurs potentiels :



En réalité...



... L'appelé n'interagit pas directement avec le téléphone public.

Diagramme de contexte statique

Diagramme de cas d'utilisation

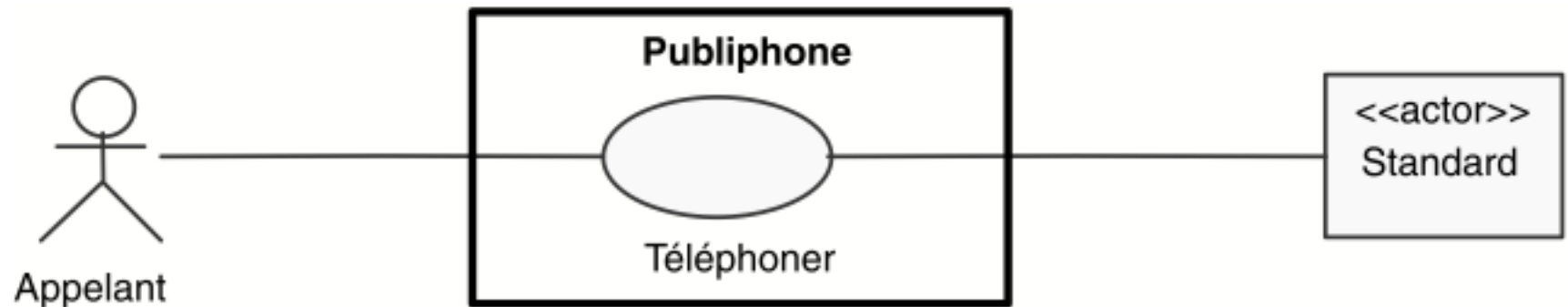


Diagramme de séquence système ?

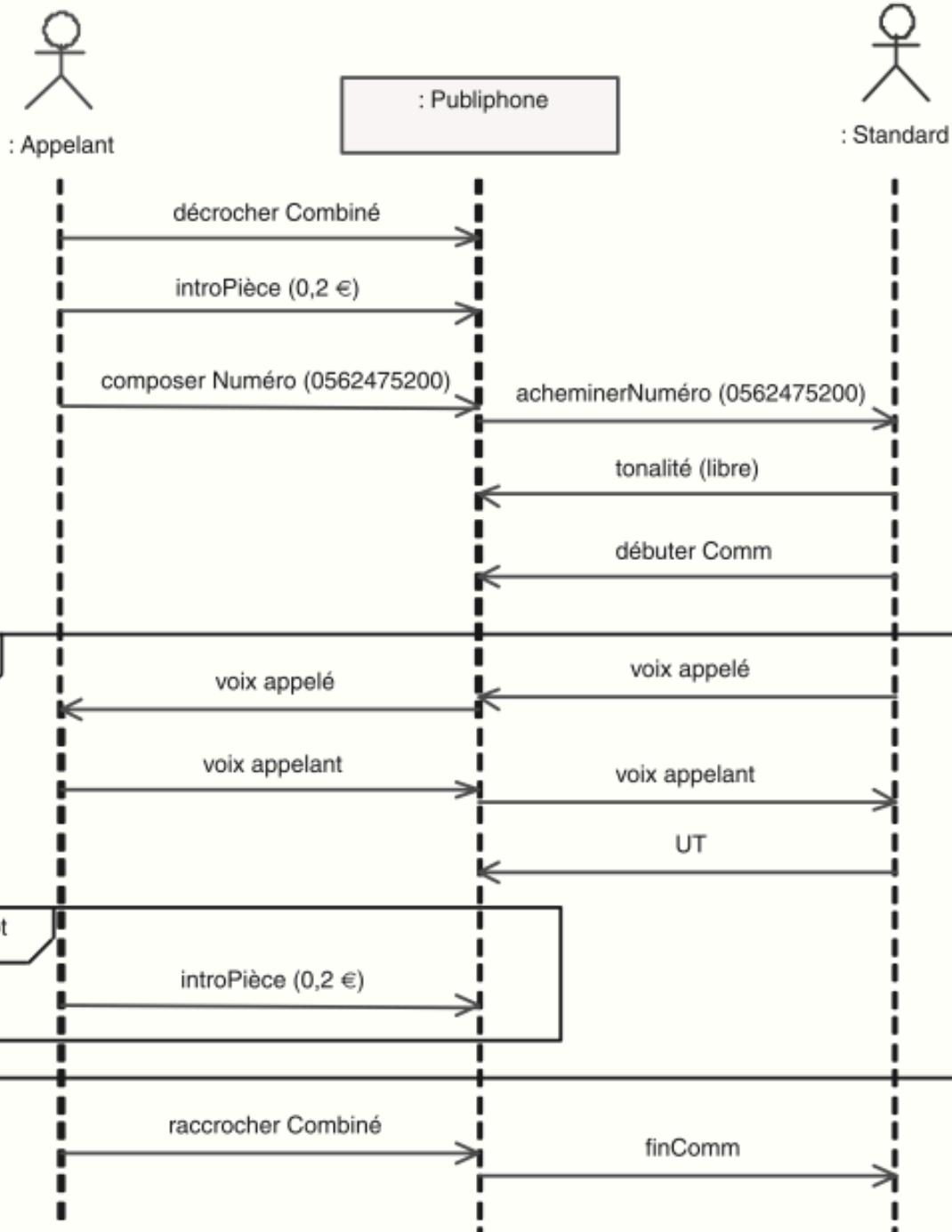


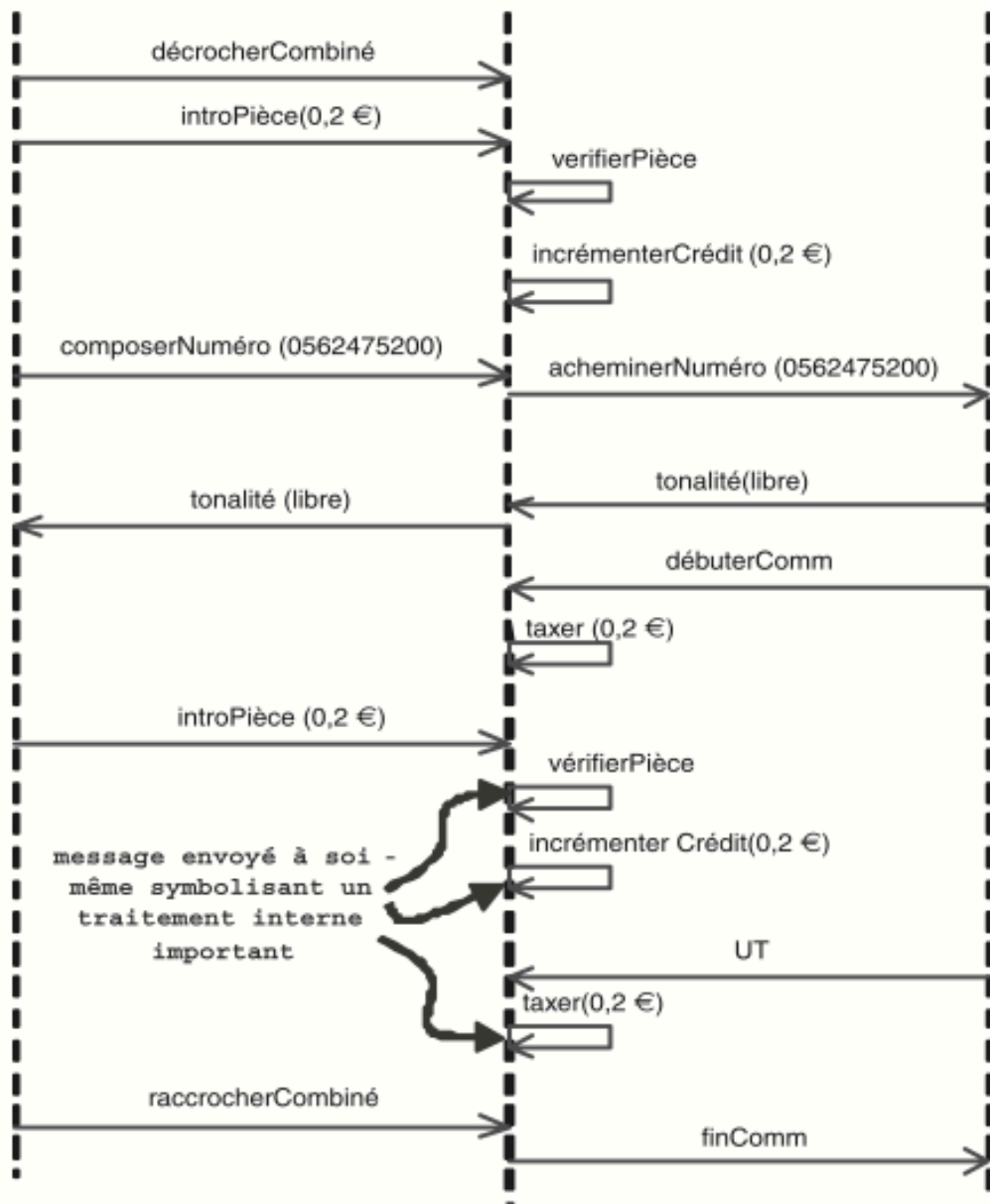
Diagramme de séquence système **enrichi** ?

- Ajouter les activités internes du systèmes
- Uniquement les opérations du système
(retirer les activités liées à la conversation)

: Appellant

: Publiphone

: Standard



Représentation du contexte dynamique

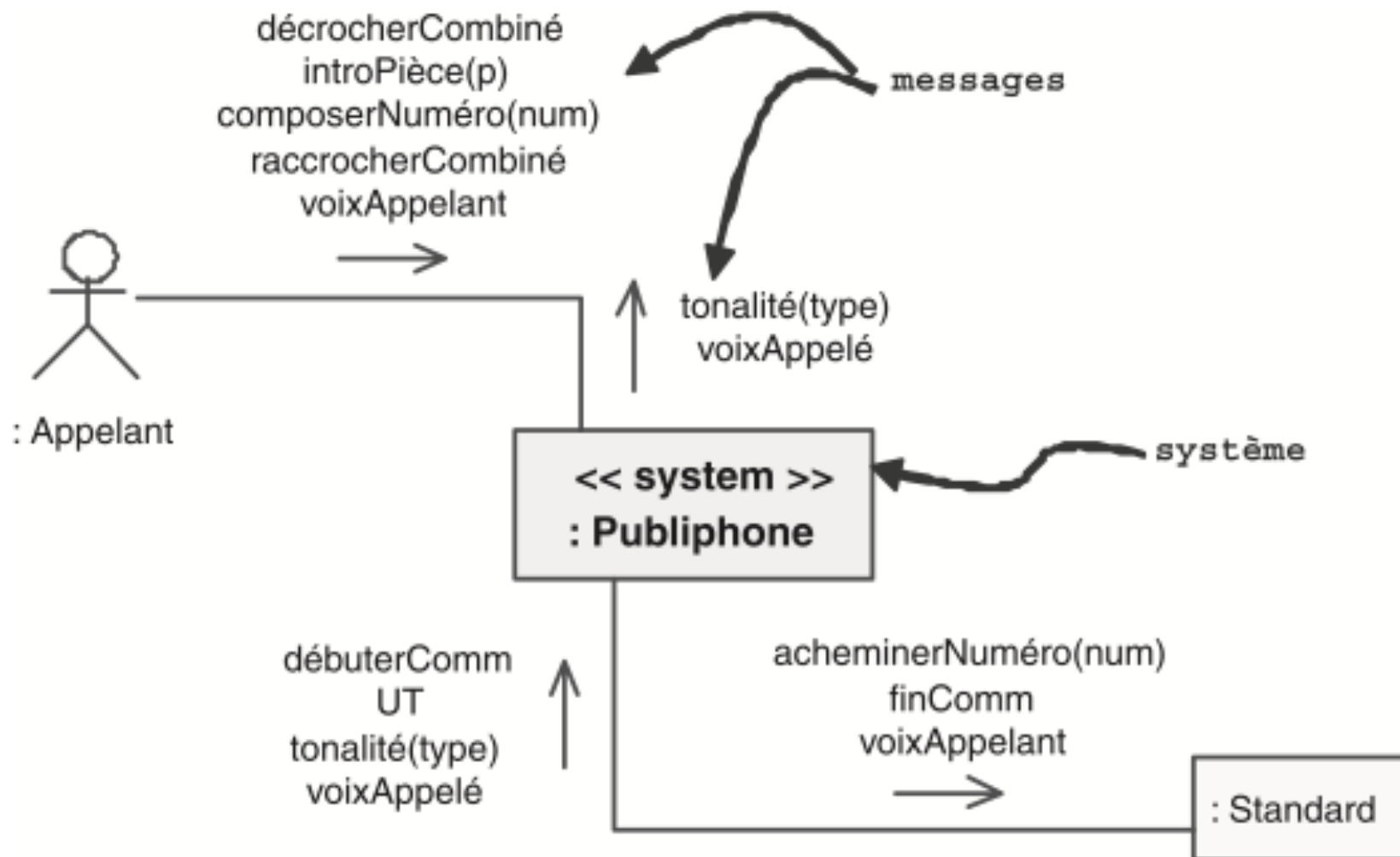
- Lister l'ensemble des messages émis et reçus par le téléphone.
 - Message reçus = évènements qui déclenches les transitions
 - Message émis = actions sur les transitions
- Utilisation d'un **diagramme de contexte dynamique** pour représenter les messages échangés

Diagramme de contexte dynamique

- Le système est un **objet au centre** du diagramme
- Objet entouré par une **instance de chaque acteur**
- Un lien relie le système à chaque acteur
- Sur chaque lien, **tous les messages** (entrées et sorties du système) sont répertoriés

Réaliser le diagramme de contexte dynamique

- A partir des diagrammes de séquence système



Enrichissez le diagramme

- Sur le diagramme système : **uniquement le scénario nominal**
- Il faut rajouter :
 - Les pièces à rendre par le téléphone
 - L'envoi d'un signal du téléphone vers le standard pour déclencher le décompte du temps
 - En cas de numéro non valide
 - Si l'appelé raccroche le premier
 - L'état de la ligne (pas uniquement le type de tonalité)

- Nombreux messages dans le diagramme de contexte dynamique = comportement non trivial
 - ➔ Construction incrémentale et itérative !

Réaliser un diagramme d'état simple

- Description du comportement **nominal** du téléphone

Diagramme d'état simple

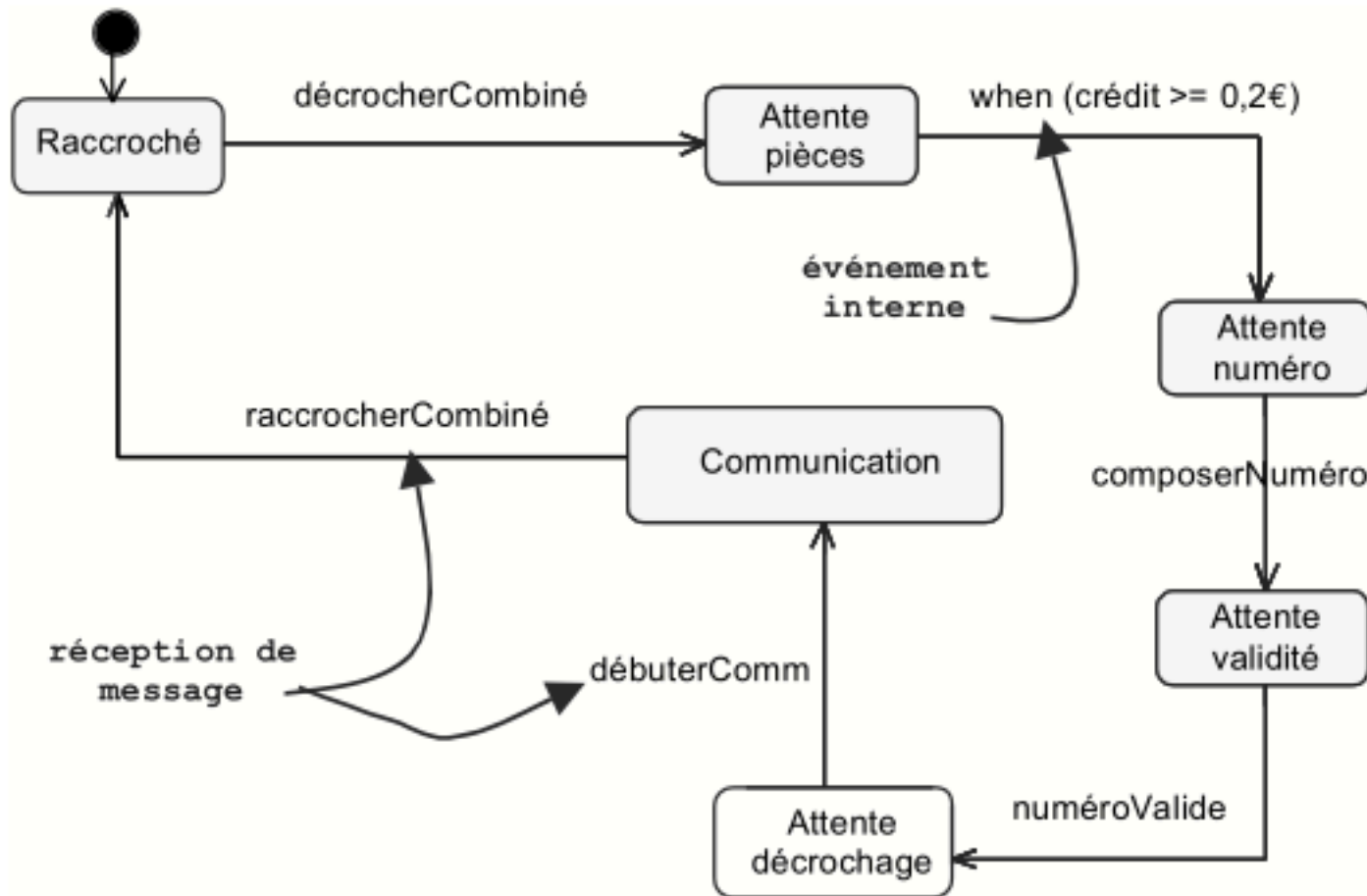
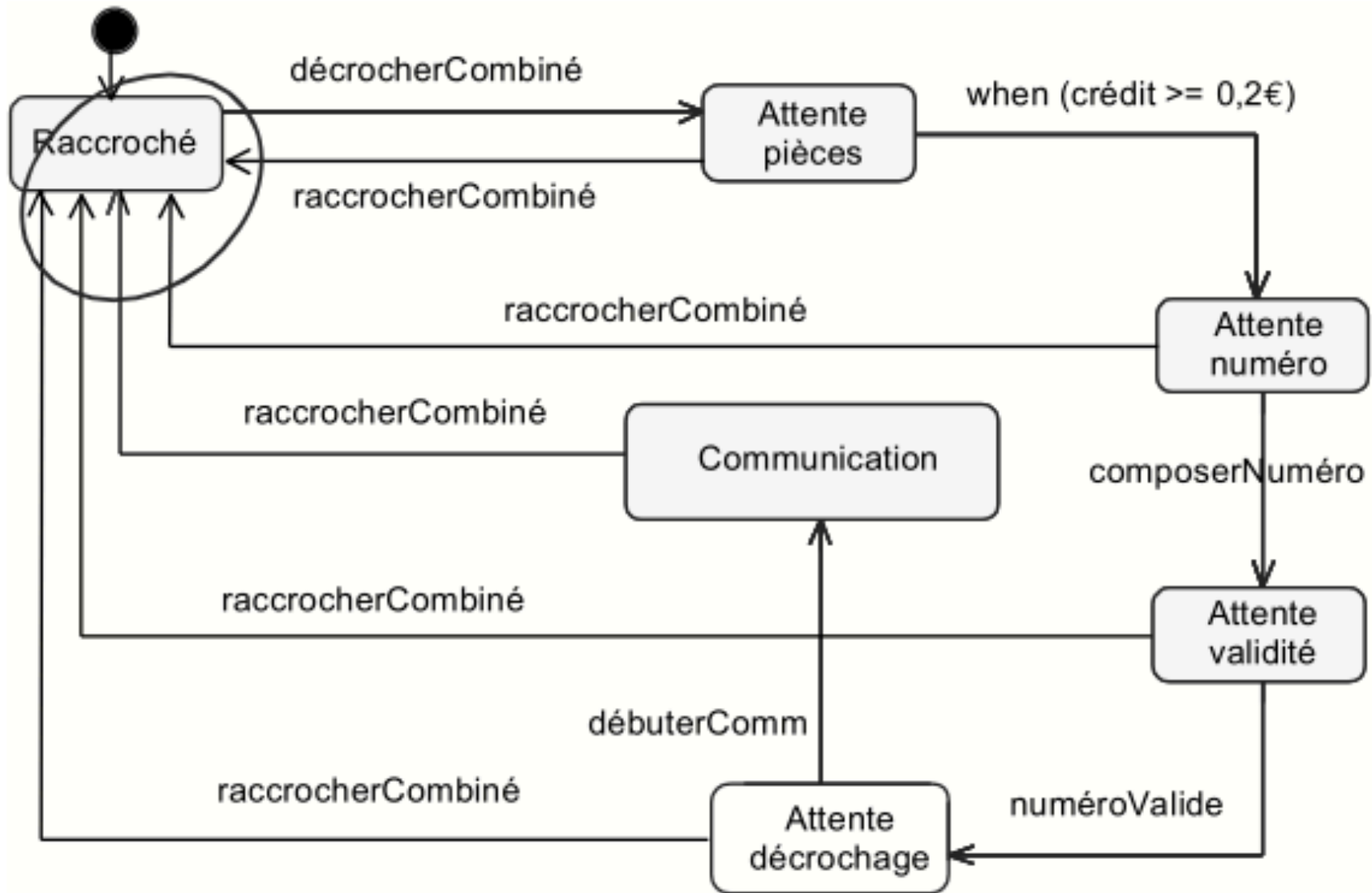


Diagramme d'états

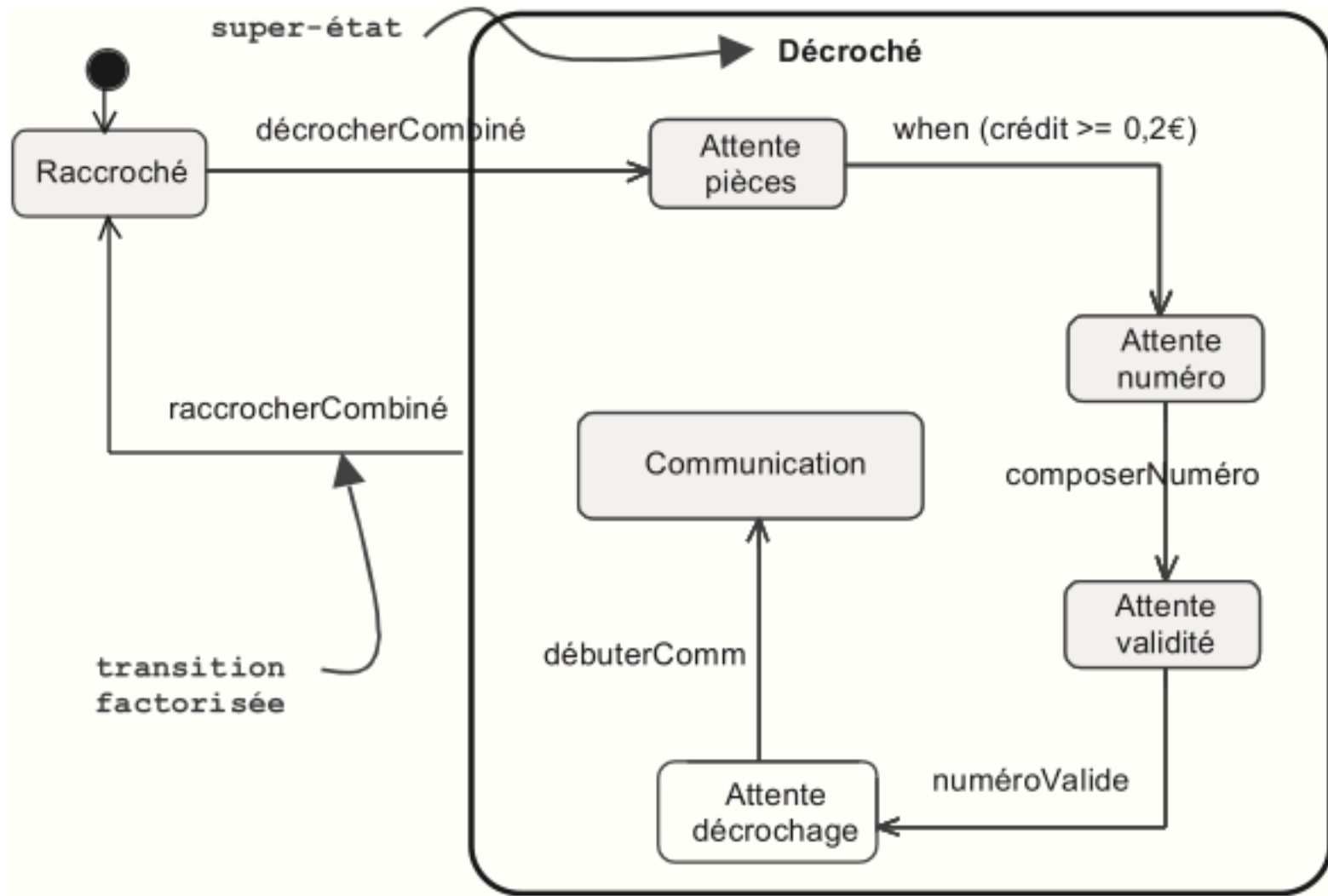
- L'utilisateur peut raccrocher à tout moment : comment le représenter ?

Solution triviale



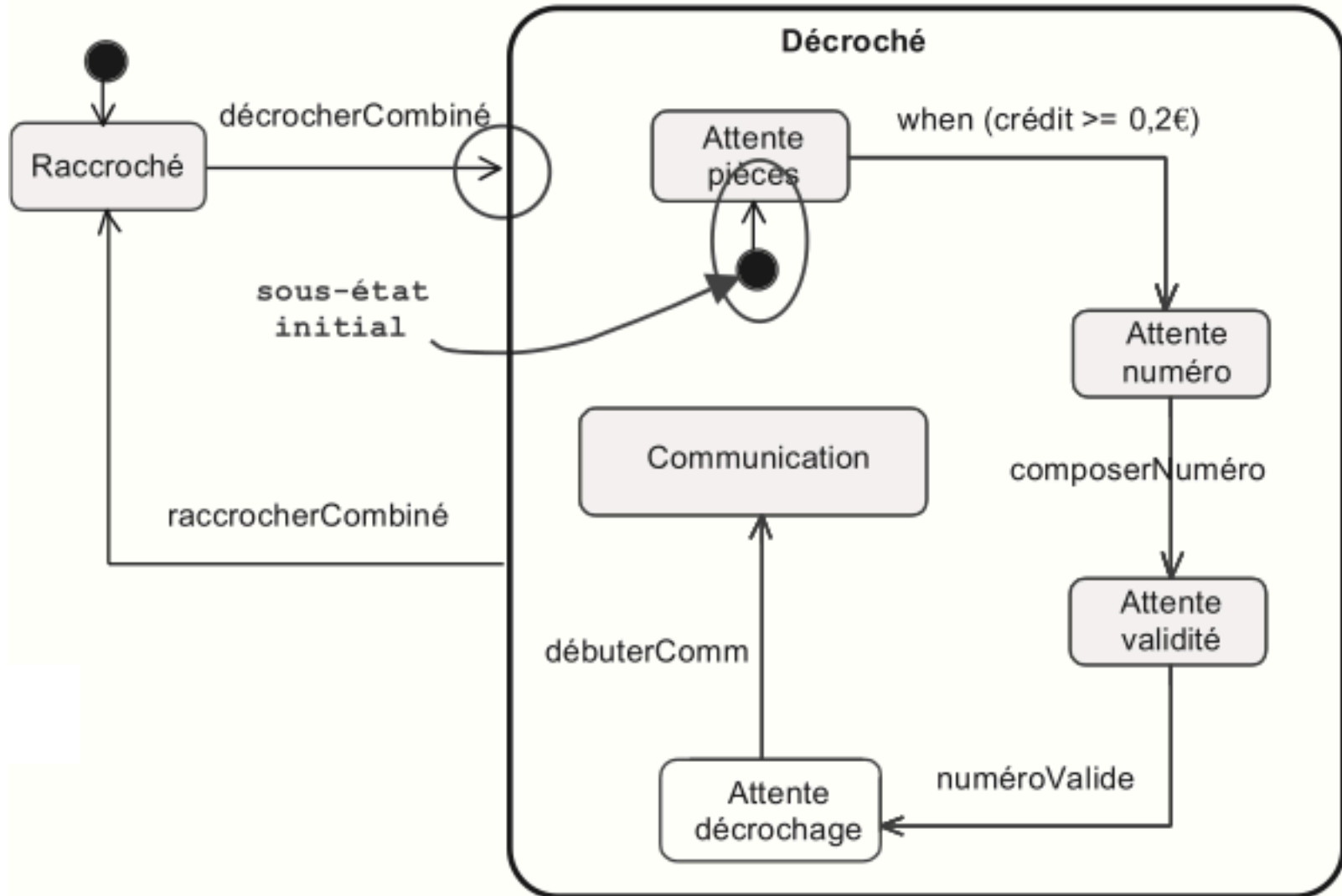
Des transitions déclenchées par l'événement raccrocher Combiné qui sortent de tous les états

Solution plus élégante



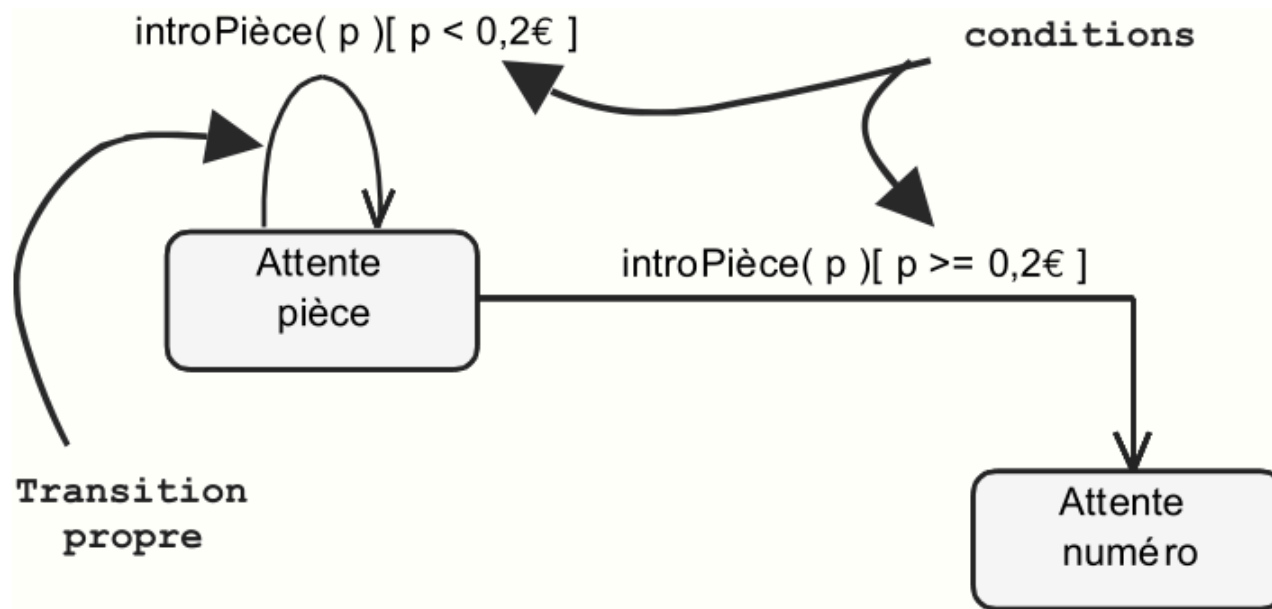
Super-état (état composite) « Décroché » pour factoriser la transition de sortie vers l'état « Raccroché »

Variante avec sous-état initial



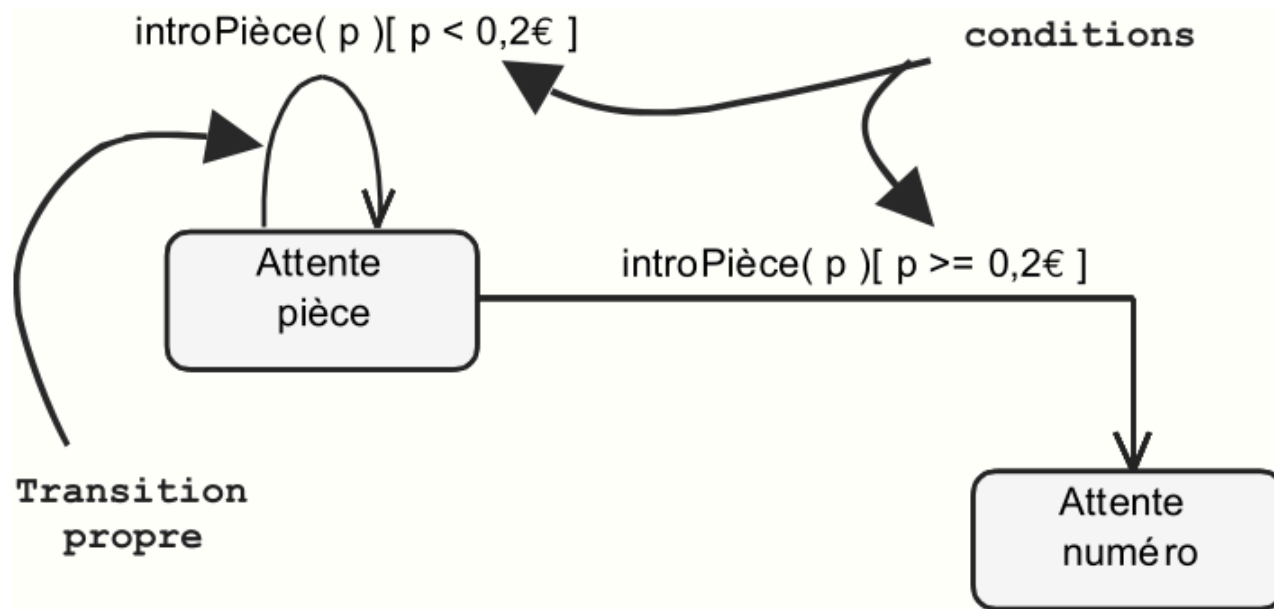
- Comment prendre en compte les pièces introduite par l'appelant ?
 - On teste la condition crédit $> 0,2\text{€}$ associé à l'évènement interne (when)
 - Mais on ne prend pas en compte l'action de l'appelant

Solution 1



Ca ne marche pas, pourquoi ??

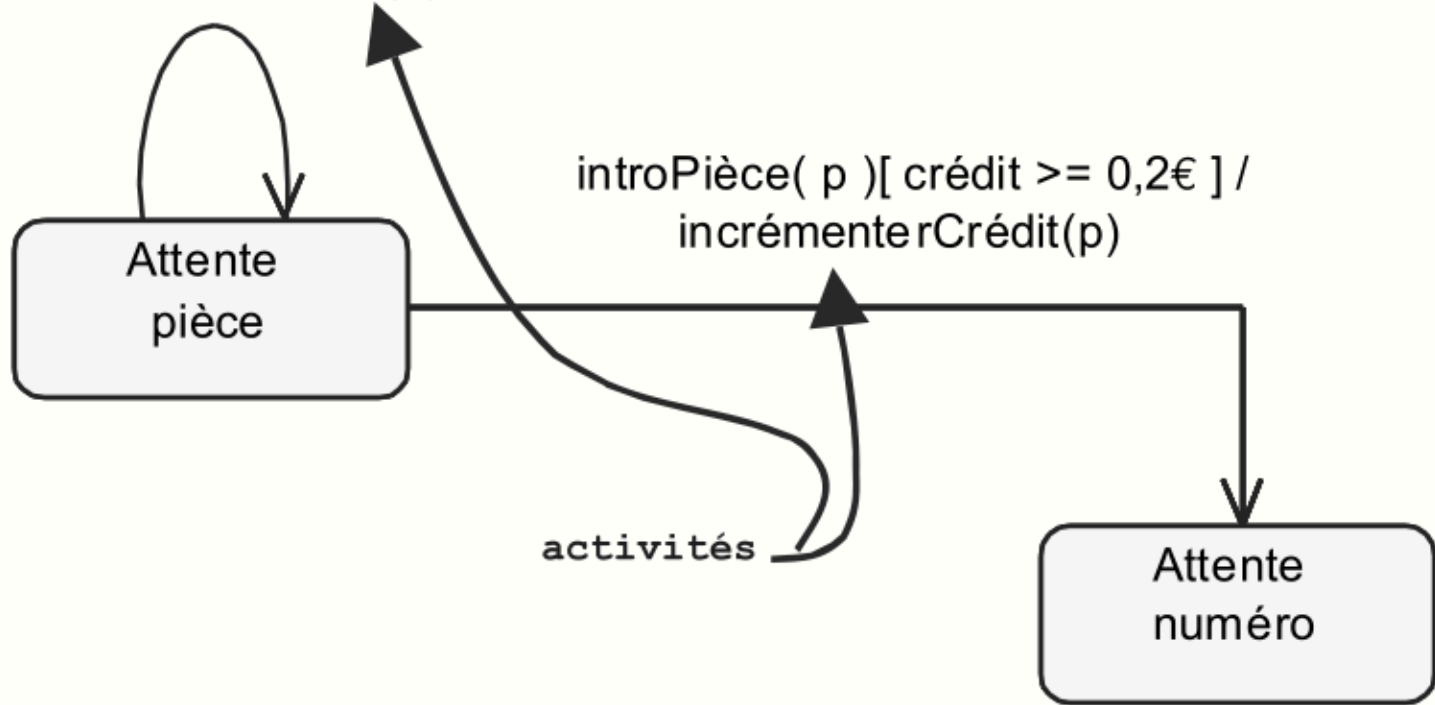
Solution 1



Si on mets 2 pièces de 10 centimes, on ne peut pas appeler : $p < 20$

Solution 2

introPièce(p) [crédit < 0,2€]
/ incrémenterCrédit(p)

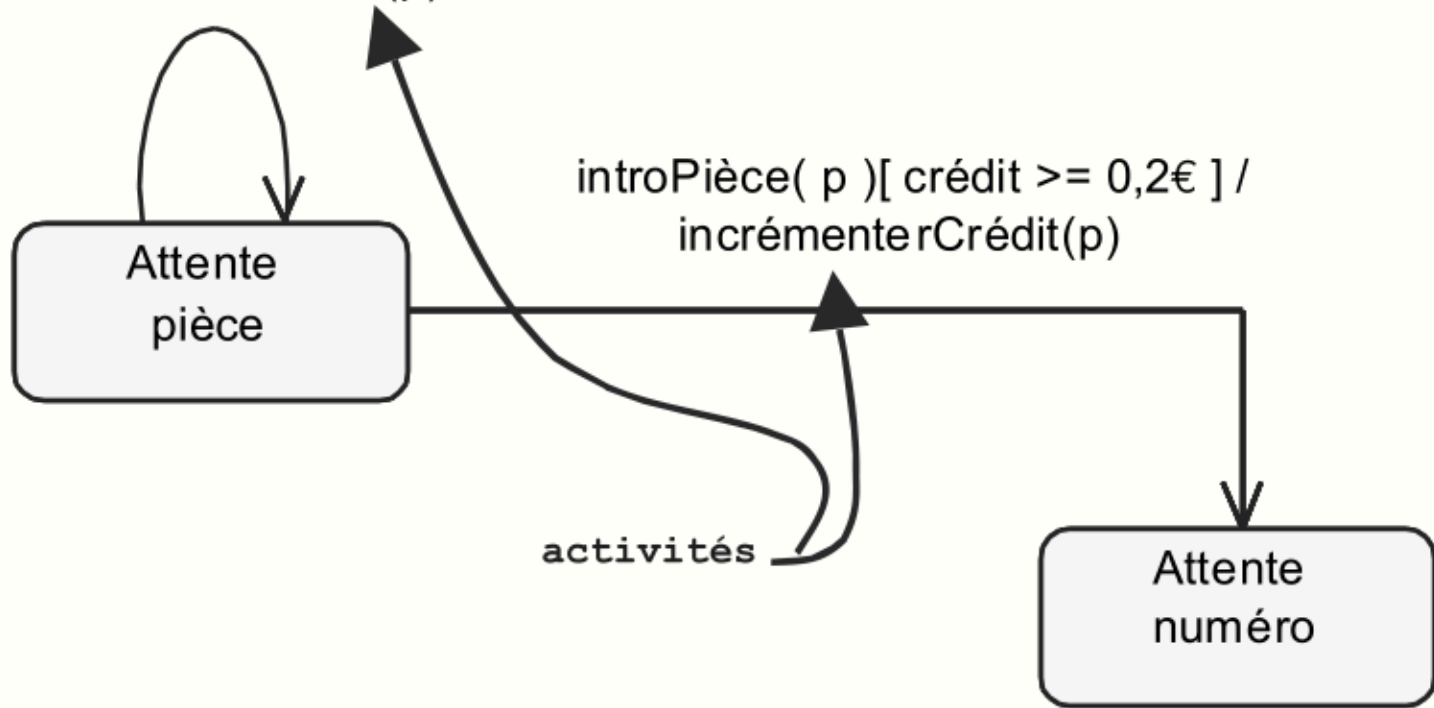


C'est mieux, mais toujours pas bon !!

POURQUOI ??

Solution 2

introPièce(p) [crédit < 0,2€]
/ incrémenterCrédit(p)



L'effet est appliqué **APRES** le test de la condition !!

1^{ère} pièce de 10 ct : crédit < 0,2 → le crédit est incrémenté (crédit=0,1)

2^{ème} pièce de 10 ct : crédit < 0,2 → le crédit est incrémenté (crédit=0,2)

Solution 3

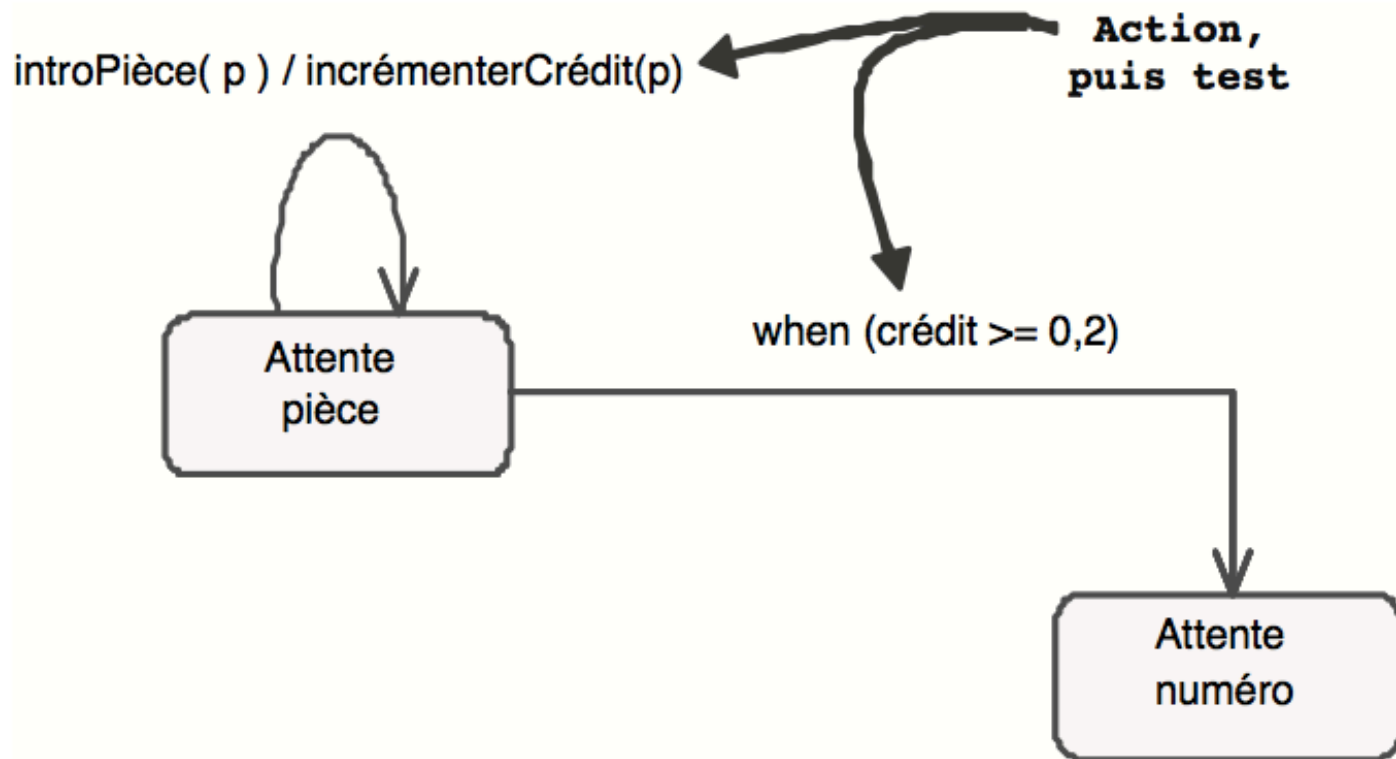


Diagramme d'état

- Comment prendre en compte la gestion complète du crédit de l'appelant ?

- Comment prendre en compte la gestion complète du crédit de l'appelant ?
 - Au décrochage : le crédit doit être initialisé à 0
 - Transition propre sur l'état attente pièce : OK
 - Transition vers Attente numéro : OK
 - Au décrochage : le crédit est débité
 - En communication : le crédit est déduit régulièrement (à chaque UT)
 - La monnaie est rendue lorsque l'appelant raccroche

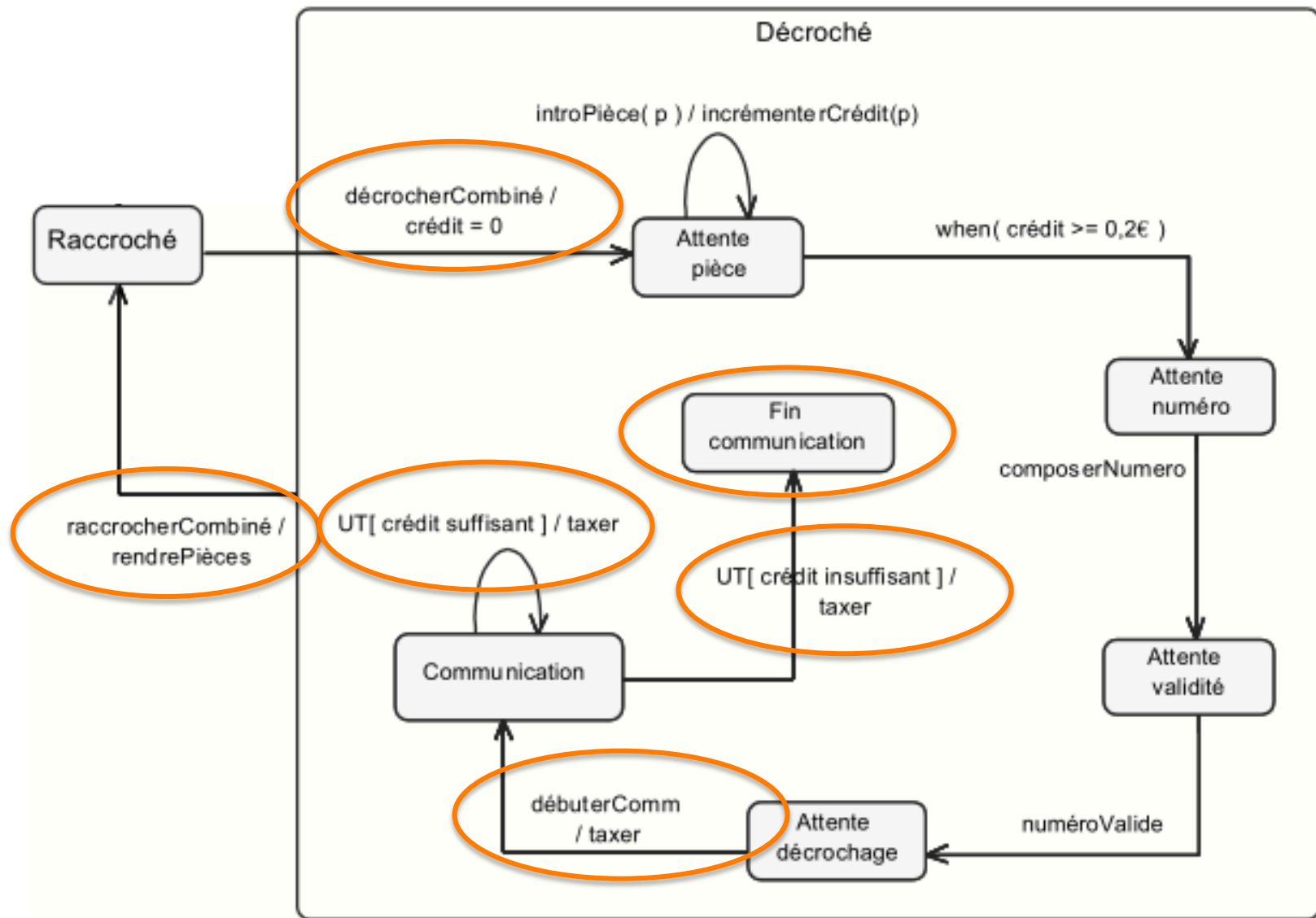
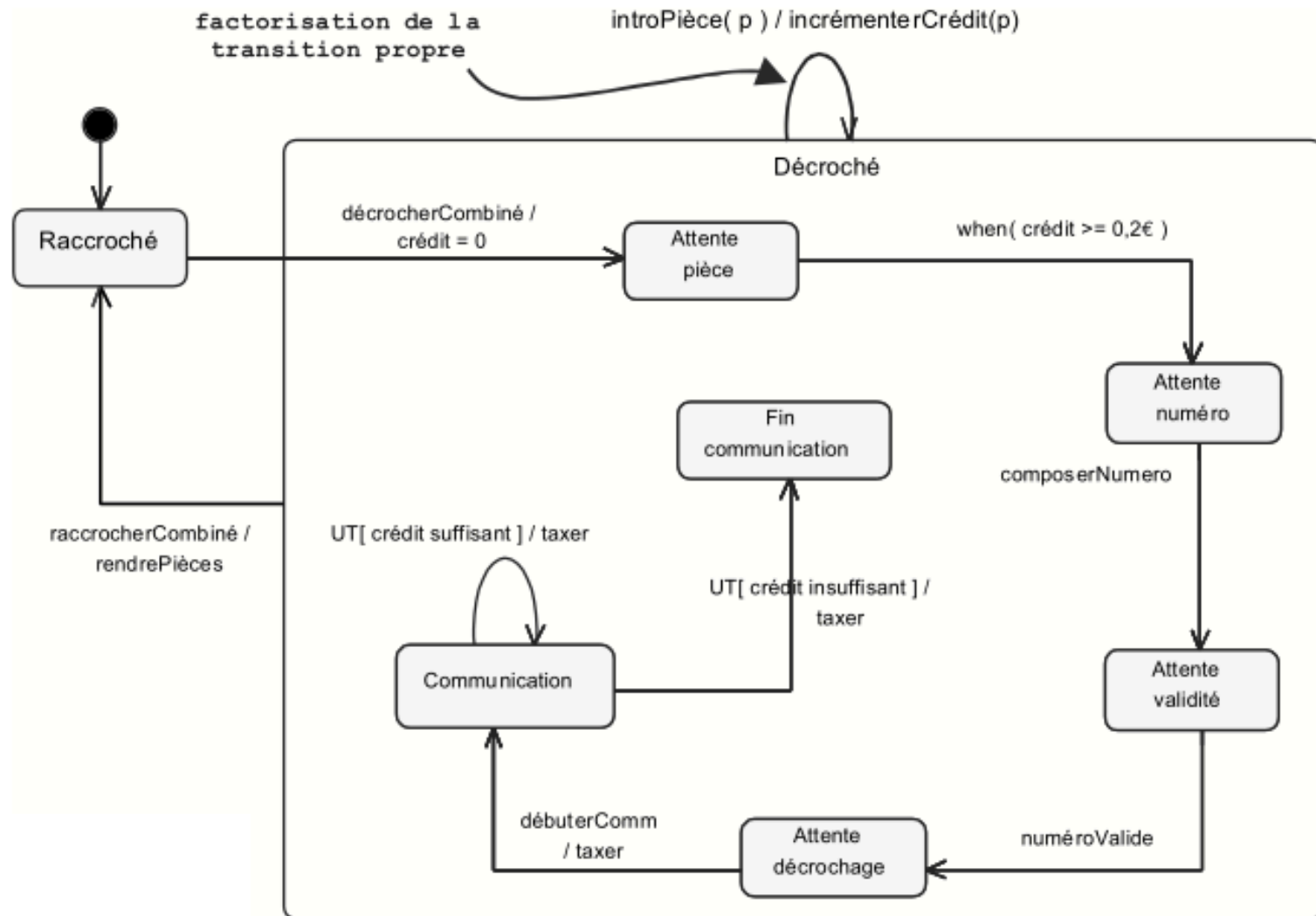


Diagramme d'état

- L'appelant peut introduire des pièces à tout moment

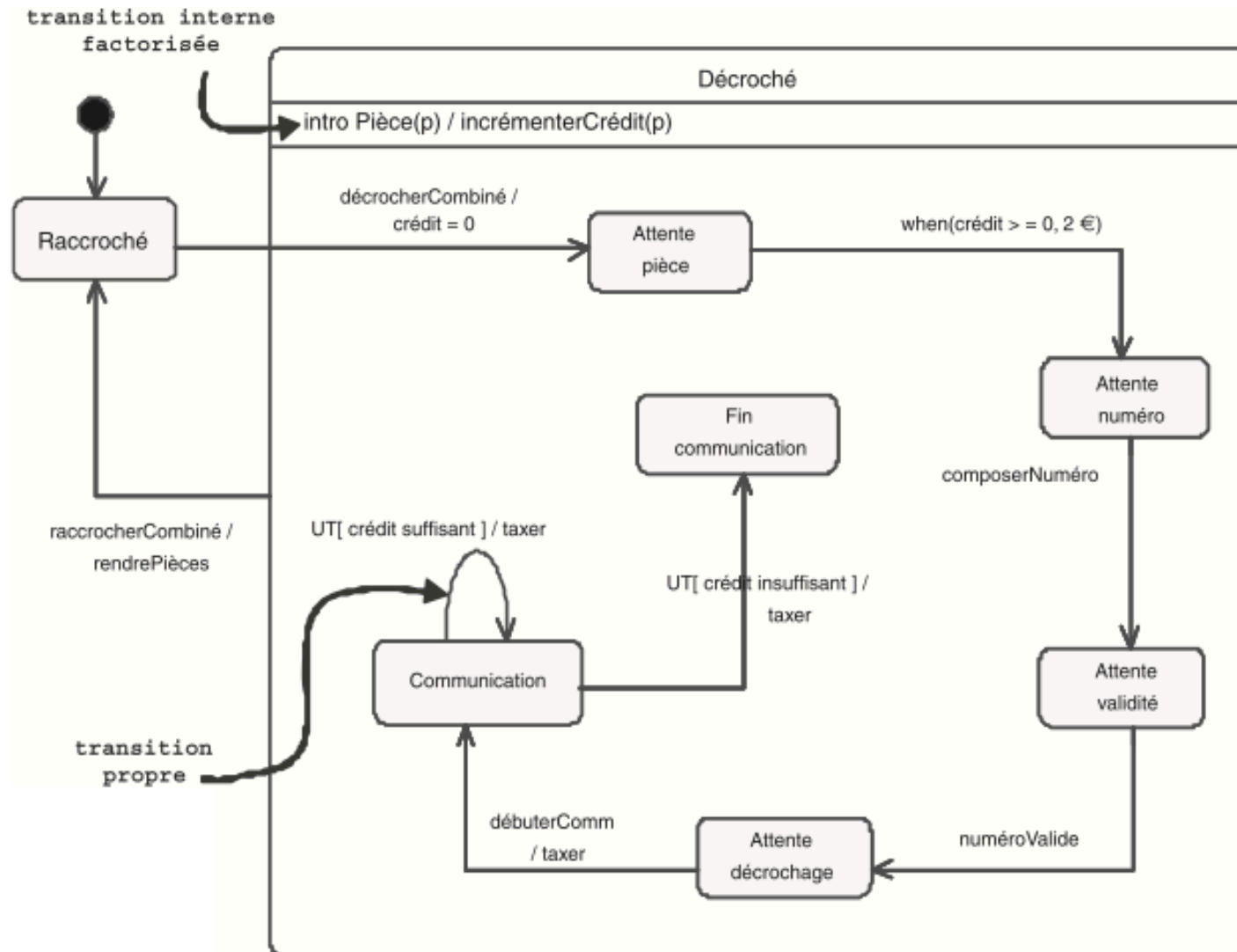
- L'appelant peut introduire des pièces à tout moment
 - Solution triviale : transition propre sur chaque état
→ très lourd
 - Solution plus élégante : les **transitions internes**
→ couple événement / effet MAIS sans influence sur l'état courant
 - \neq **transitions propres** : on sort de l'état et on retourne dans le même état (et implicitement au sous état initial)

Exemple de transition propre



Problème : retour au sous état initial « attente pièce »

Avec une transition interne



- L'appelant à deux minutes pour composer son numéro après l'introduction de la monnaie
 - *timerNumérotation* envoyé par le téléphone au standard
 - *timeOutNumérotation* envoyé par le standard au téléphone
- En UML, envoi de message : **send**
 - Envoi d'un message à un autre objet pendant une transition
 - `send cible.message`

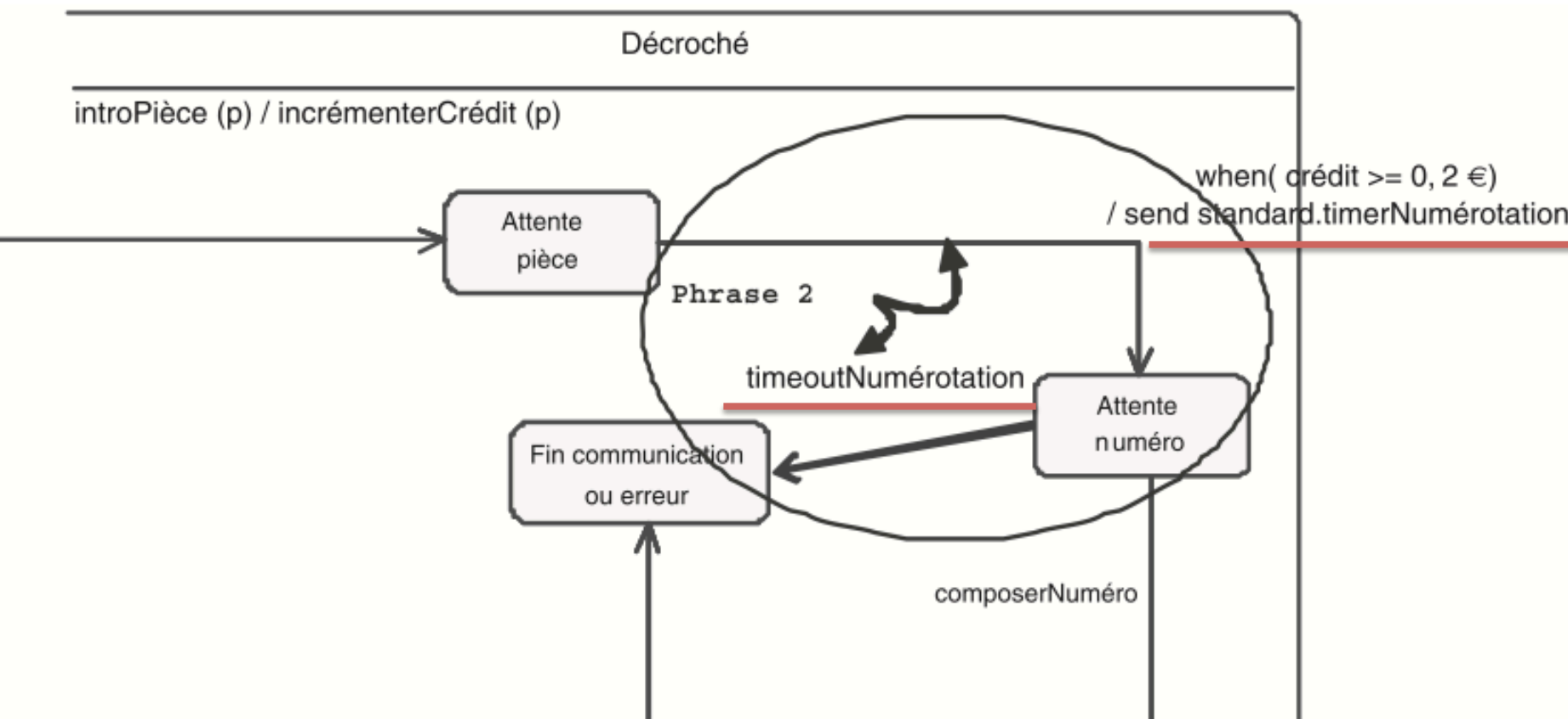


Diagramme d'état final

