

Algèbres de processus pour la substitutivité et la compatibilité de services en ligne

VOINOT Jérôme

ARA COPS

Toulouse - 5 octobre 2006

Plan de la présentation

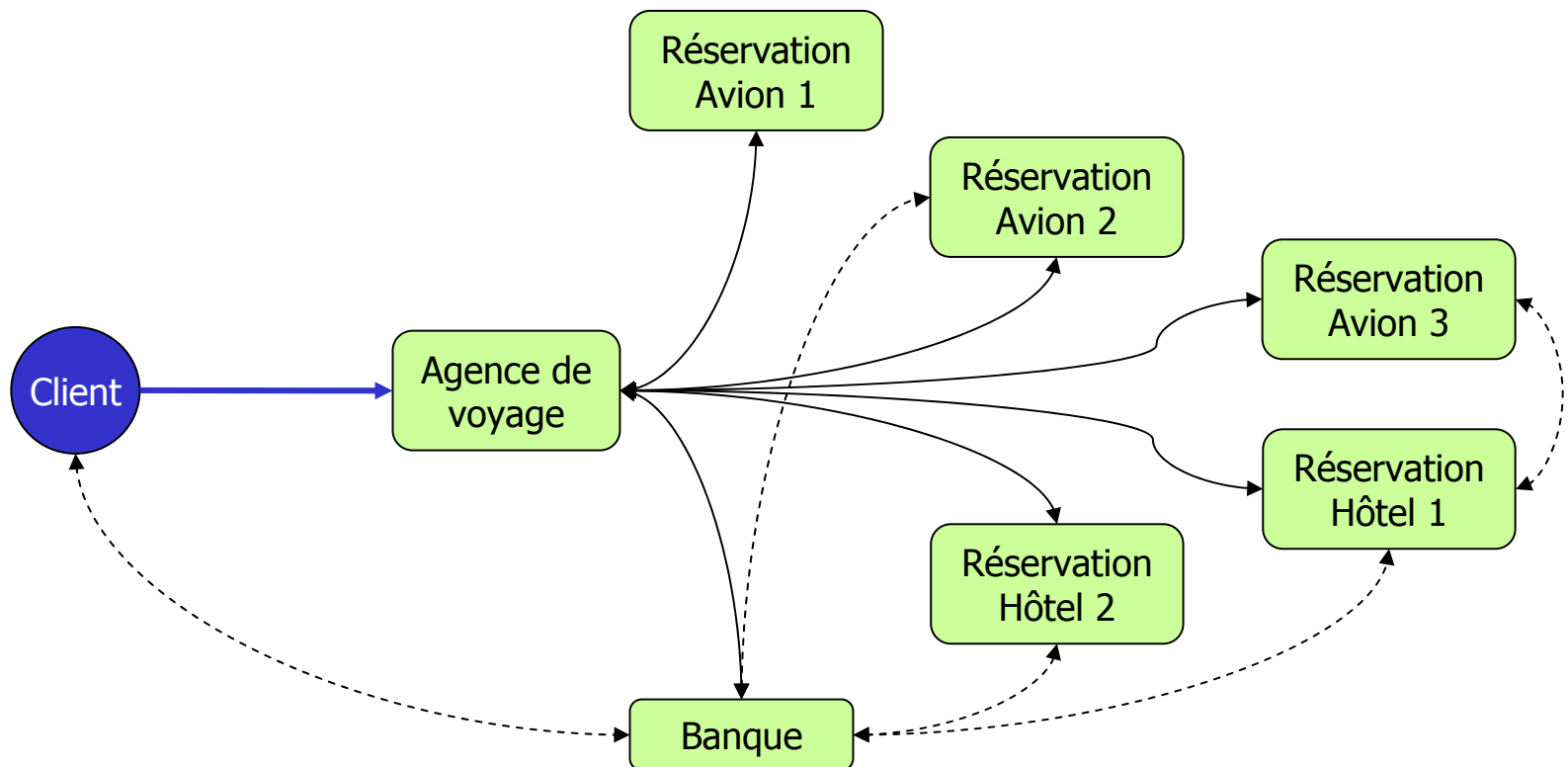
- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

Plan de la présentation

- **Motivations**
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

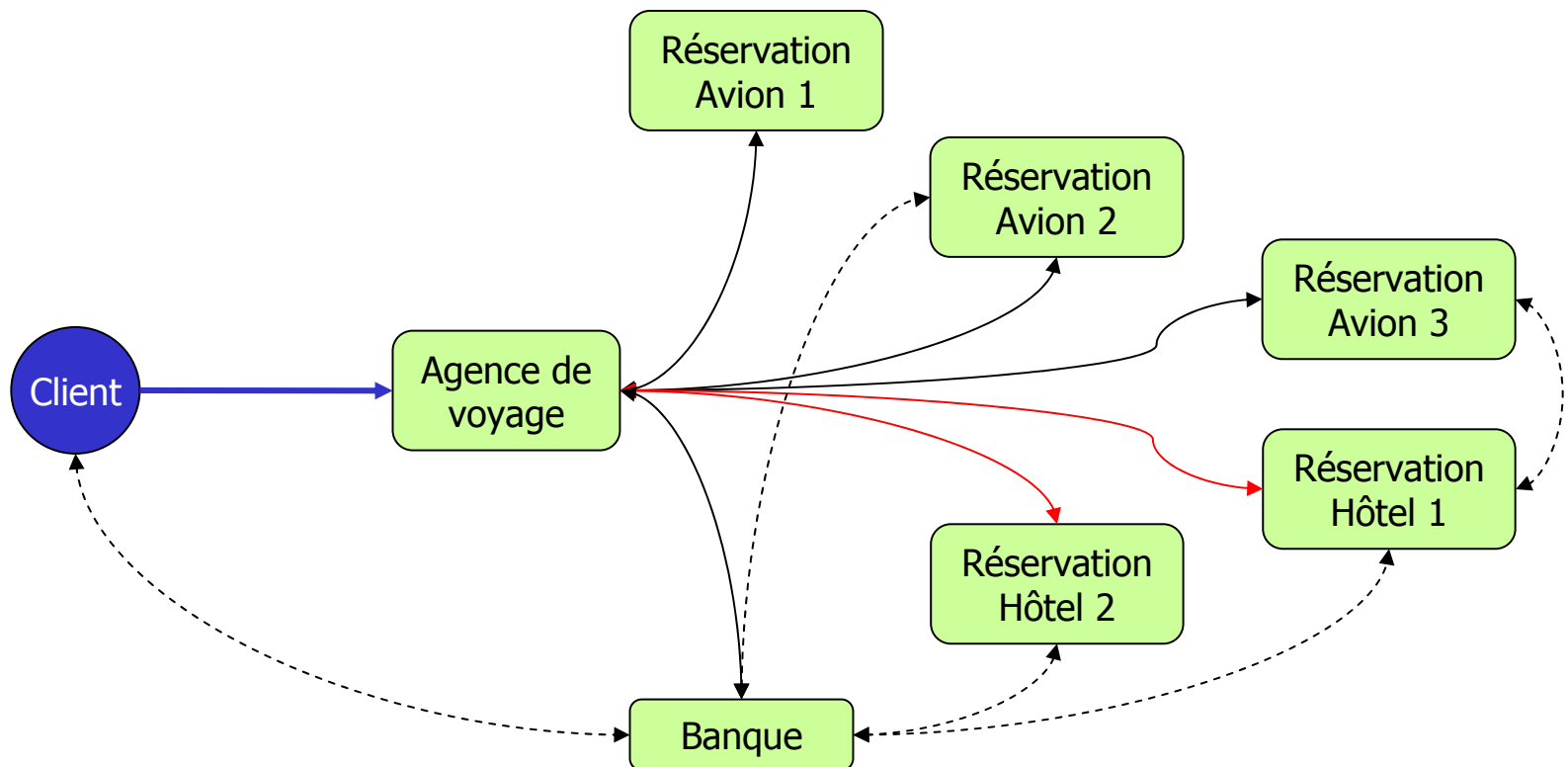
Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



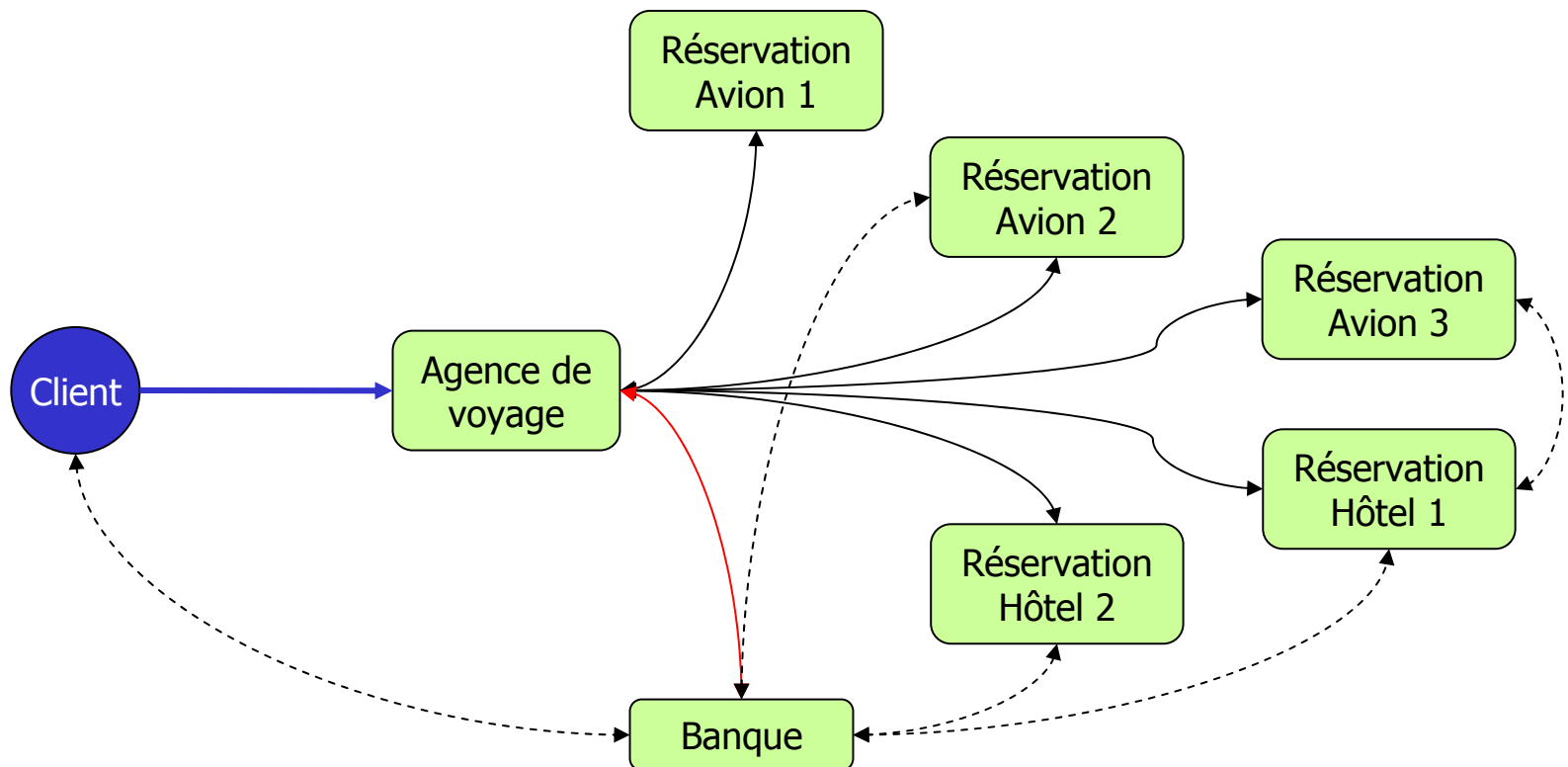
Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



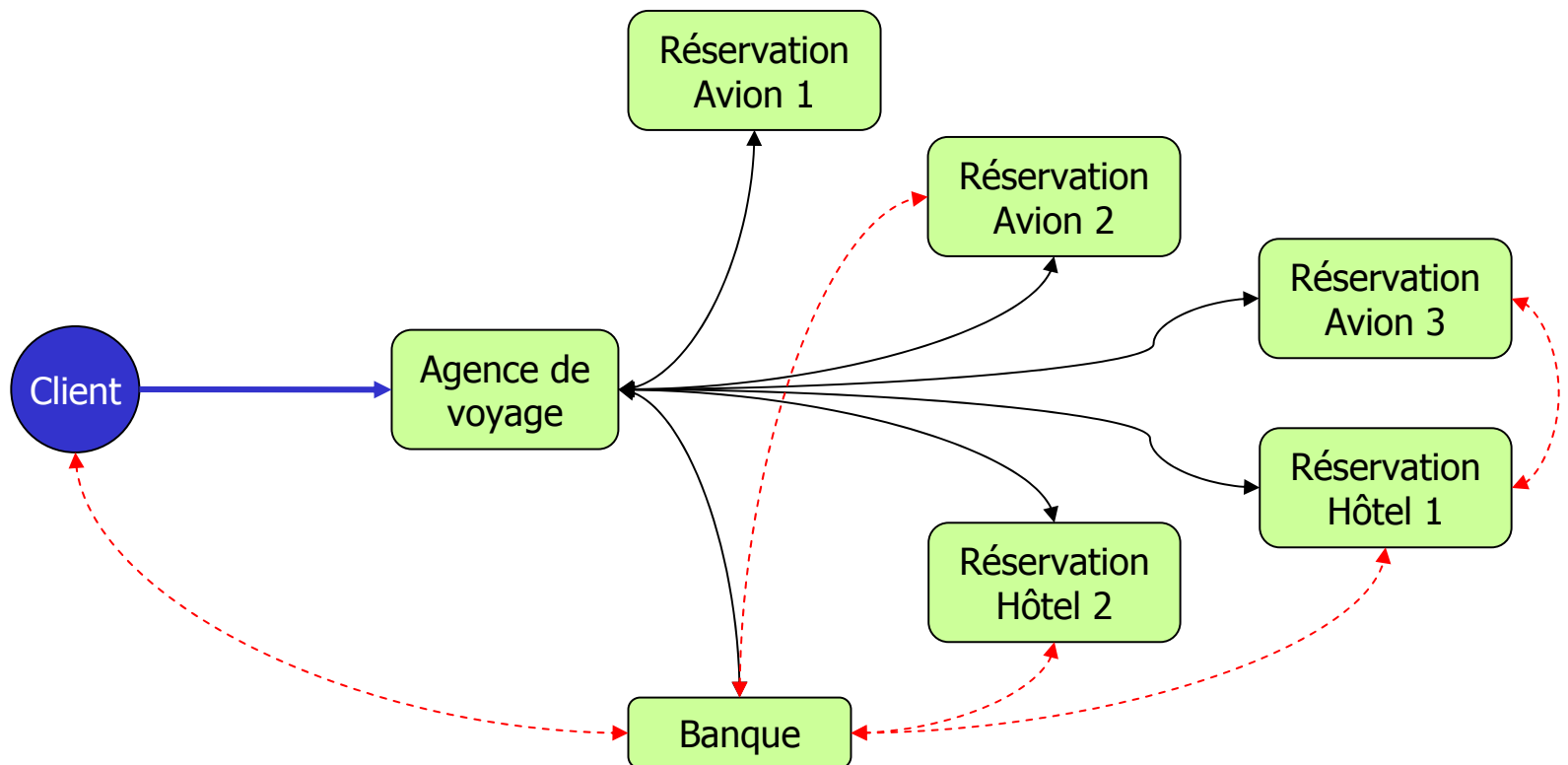
Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



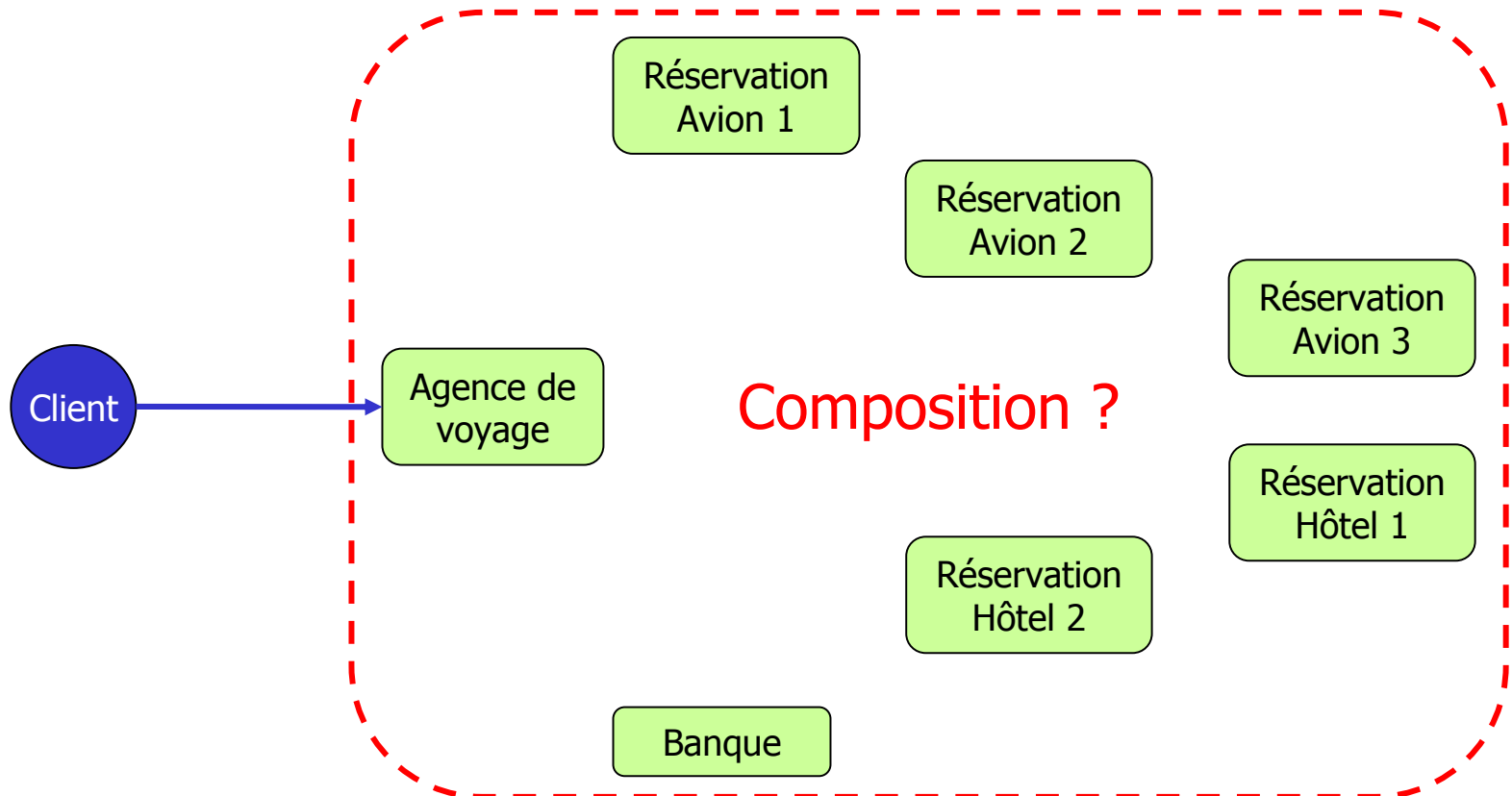
Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



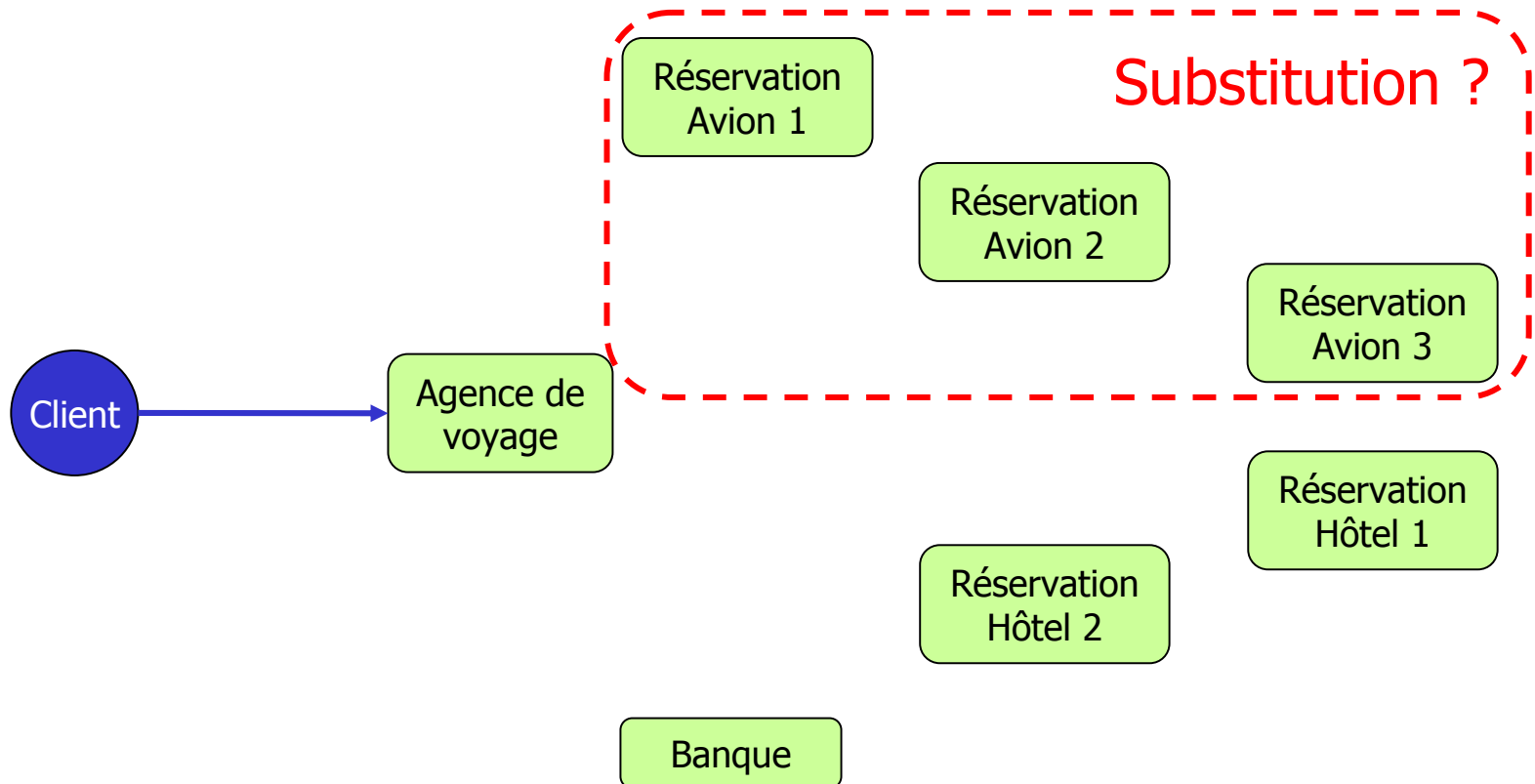
Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



Motivations

- Utilisation des services en ligne en tant que composants d'applications en respectant différentes règles.



Motivations

- Tenir compte de critères quantitatifs ou qualitatifs dans le cadre de la composition de services en ligne.

Motivations

- Tenir compte de critères quantitatifs ou qualitatifs dans le cadre de la composition de services en ligne.
 - Ne pas substituer un service par un autre si ces deux services n'ont pas un niveau de confiance similaire.

Motivations

- Tenir compte de critères quantitatifs ou qualitatifs dans le cadre de la composition de services en ligne.
 - Ne pas substituer un service par un autre si ces deux services n'ont pas un niveau de confiance similaire.
 - Réaliser une composition en tenant compte du temps d'exécution globale, ou encore de son coût financier.

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

Algèbres de processus et services en ligne

- Utilisation de l'algèbre de processus CCS avec passage de valeurs [Mil89].

Algèbres de processus et services en ligne

- Utilisation de l'algèbre de processus CCS avec passage de valeurs [Mil89].
- Syntaxe de CCS avec passage de valeurs :

$$P, Q ::= A(e_1, \dots, e_n) \mid a(x).P \mid \bar{a}(e).P \mid \tau.P \mid \sum_{i \in I} P_i \mid \\ P \mid Q \mid P[f] \mid P \setminus L \mid \text{if } b \text{ then } P \text{ else } Q$$

Algèbres de processus et services en ligne

- Sémantique de CCS avec passage de valeurs :

$$\text{Act}_{\text{in}} \quad \frac{}{a(x).P \xrightarrow{a(n)} P[n/x]} \quad (n \geq 0)$$

$$\text{Act}_{\text{out}} \quad \frac{}{\bar{a}(e).P \xrightarrow{\bar{a}(n)} P} \quad (n \text{ est le résultat de l'évaluation de } e)$$

Algèbres de processus et services en ligne

- Sémantique de CCS avec passage de valeurs :

$$\text{Act}_{\text{in}} \quad \frac{}{a(x).P \xrightarrow{a(n)} P[n/x]} \quad (n \geq 0)$$

$$\text{Act}_{\text{out}} \quad \frac{}{\bar{a}(e).P \xrightarrow{\bar{a}(n)} P} \quad (n \text{ est le résultat de l'évaluation de } e)$$

- Intégration d'une notion de coût sur les transitions.

Pourquoi utiliser l'algèbre de processus CCS ?

- Modélisation de l'ensemble des mécanismes liés aux services en ligne et à leurs compositions :

Pourquoi utiliser l'algèbre de processus CCS ?

- Modélisation de l'ensemble des mécanismes liés aux services en ligne et à leurs compositions :
 - [SBS04] : utilisation de CCS, vérification de compatibilité et de la substitutivité de services en ligne.

Pourquoi utiliser l'algèbre de processus CCS ?

- Modélisation de l'ensemble des mécanismes liés aux services en ligne et à leurs compositions :
 - [SBS04] : utilisation de CCS, vérification de compatibilité et de la substitutivité de services en ligne.
 - [BCPV04,CCCV05] : utilisation de CCS avec passage de valeurs, vérification de la compatibilité et de la substitutivité de services en ligne.

Pourquoi utiliser l'algèbre de processus CCS ?

- Résultats de décidabilité pour certains fragments :

Pourquoi utiliser l'algèbre de processus CCS ?

- Résultats de décidabilité pour certains fragments :
 - [CHM93] : bisimulation pour les processus de BPP.

Pourquoi utiliser l'algèbre de processus CCS ?

- Résultats de décidabilité pour certains fragments :
 - [CHM93] : bisimulation pour les processus de BPP.
 - [Ber04] : bisimulation faible symbolique (\approx_s) pour un fragment normé de CCS avec passage de valeurs.

Pourquoi utiliser le passage de valeurs ?

- Modélisation plus fine du comportement des services en ligne et de leurs compositions.

Pourquoi utiliser le passage de valeurs ?

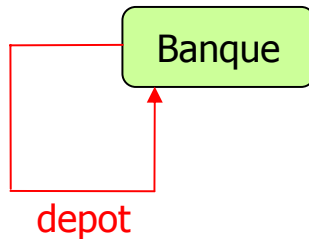
- Modélisation plus fine du comportement des services en ligne et de leurs compositions.
- [SBS04] : Impossibilité de prendre en compte les échanges de données entre services en ligne.

Pourquoi utiliser le passage de valeurs ?

Banque

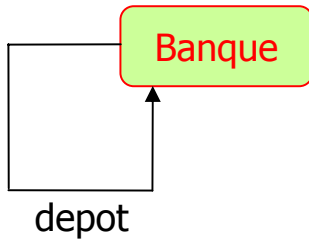
$$\text{Banque} = \text{depot.Banque} + \text{retrait.Banque} + \overline{\text{consultation.solde.Banque}}$$

Pourquoi utiliser le passage de valeurs ?



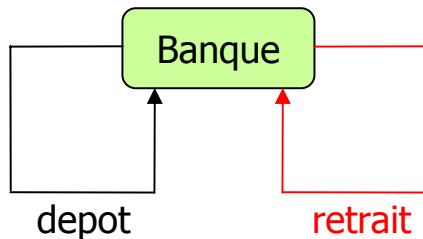
Banque = **depot**.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



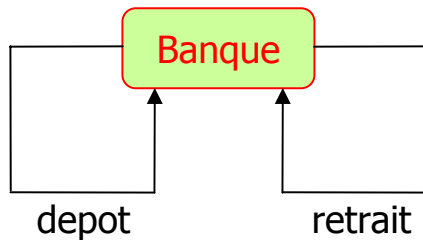
Banque = depot.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



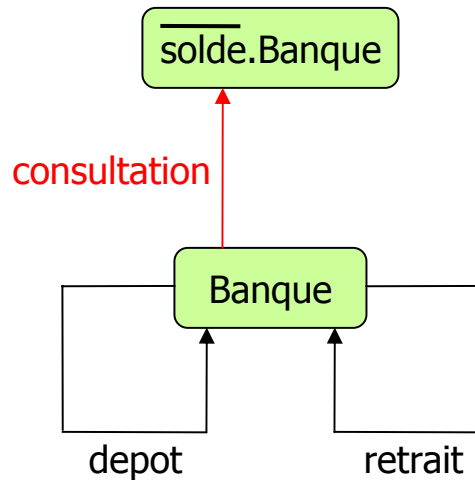
Banque = depot.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



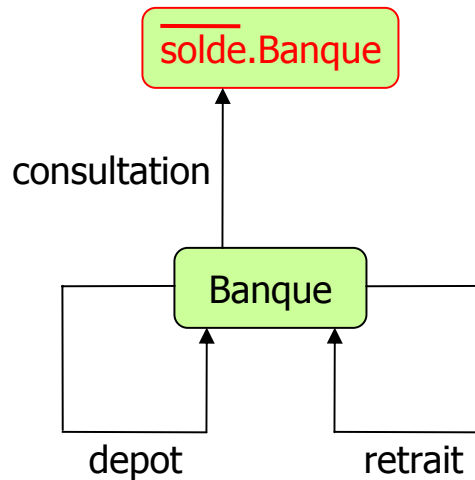
Banque = depot.Banque +
retrait.**Banque** +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



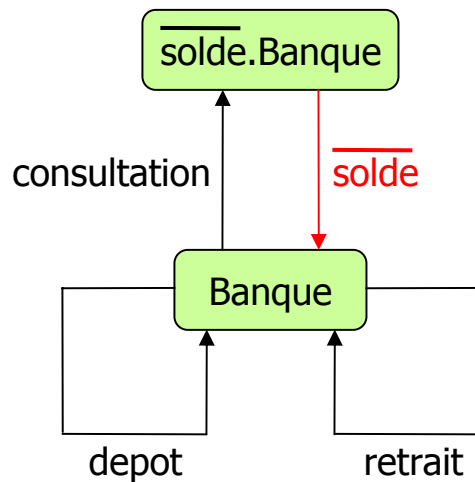
Banque = depot.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



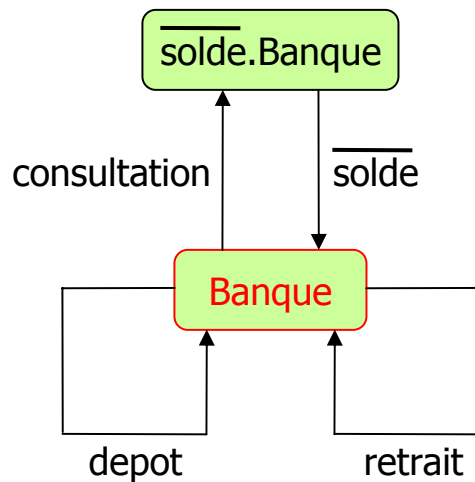
Banque = depot.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



Banque = depot.Banque +
retrait.Banque +
consultation.solde.Banque

Pourquoi utiliser le passage de valeurs ?



$$\text{Banque} = \text{depot.Banque} + \text{retrait.Banque} + \text{consultation.solde.Banque}$$

Pourquoi utiliser le passage de valeurs ?

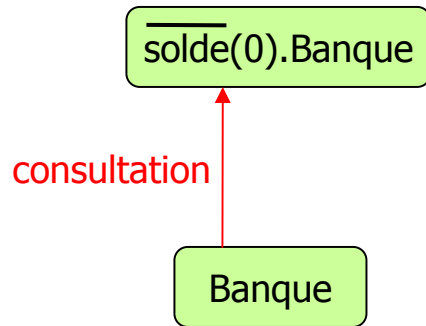
Banque

Banque = consultation.solde(0).Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation.solde(s).Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

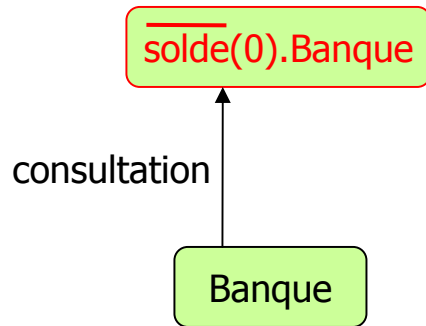


Banque = **consultation**.solde(0).Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation.solde(s).Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

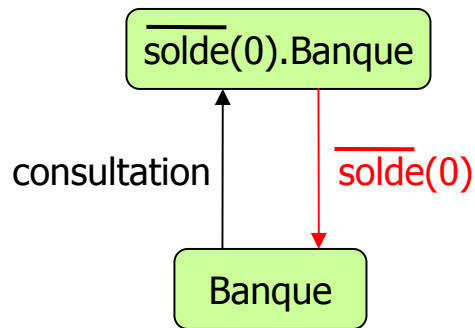


Banque = consultation.solde(0).Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation.solde(s).Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

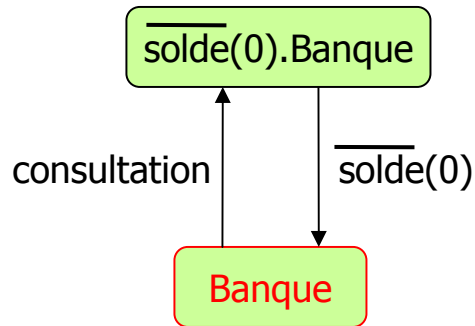


$Banque = consultation.\overline{\text{solde}(0)}.Banque + depot(d).Banque(d)$

$Banque(s) = depot(d).Banque(s + d) + consultation.\overline{\text{solde}(s)}.Banque(s) + retrait(r).Banque(s,r)$

$Banque(s,r) = \text{if } (s - r < 0) \text{ then}$
 $erreurRetrait.Banque(s)$
 $\text{else if } (s - r = 0) \text{ then}$
 $retraitOk.Banque$
 else
 $retraitOk.Banque(s - r)$

Pourquoi utiliser le passage de valeurs ?

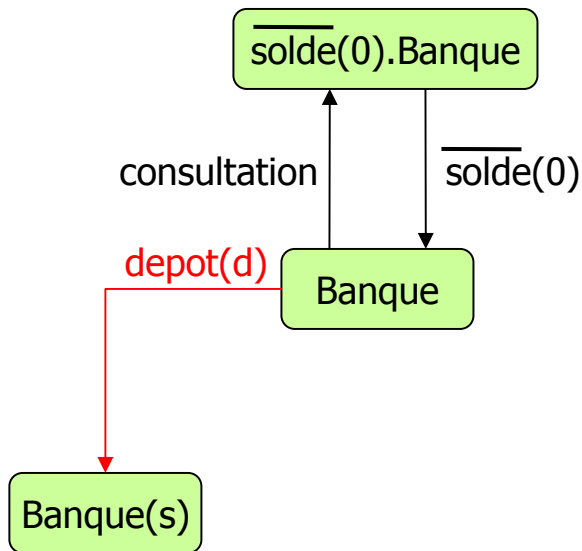


$Banque = \overline{\text{consultation}}.\overline{\text{solde}(0)}.Banque + \text{depot}(d).Banque(d)$

$Banque(s) = \text{depot}(d).Banque(s + d) + \overline{\text{consultation}}.\overline{\text{solde}(s)}.Banque(s) + \text{retrait}(r).Banque(s,r)$

$Banque(s,r) = \text{if } (s - r < 0) \text{ then}$
 $\text{erreurRetrait}.Banque(s)$
 $\text{else if } (s - r = 0) \text{ then}$
 $\text{retraitOk}.Banque$
 else
 $\text{retraitOk}.Banque(s - r)$

Pourquoi utiliser le passage de valeurs ?

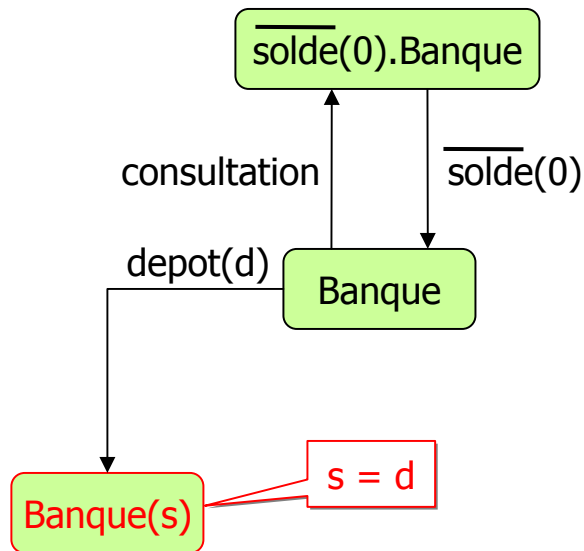


Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
 depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
 consultation. $\overline{\text{solde}(s)}$.Banque(s) +
 retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

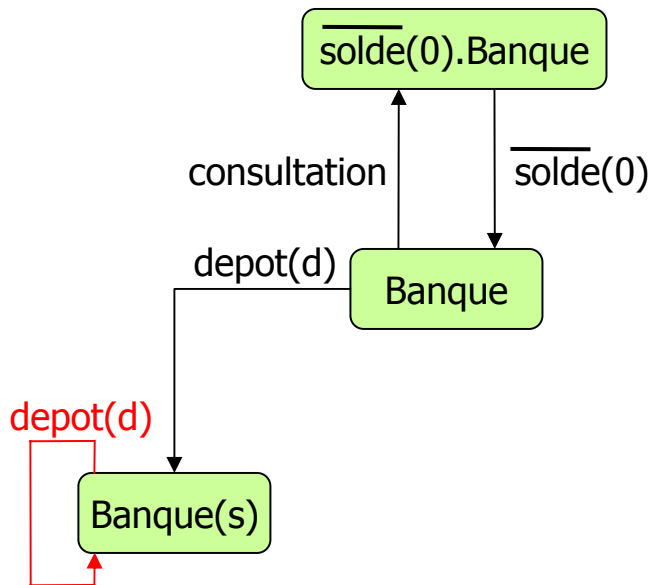


Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde}(s)}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

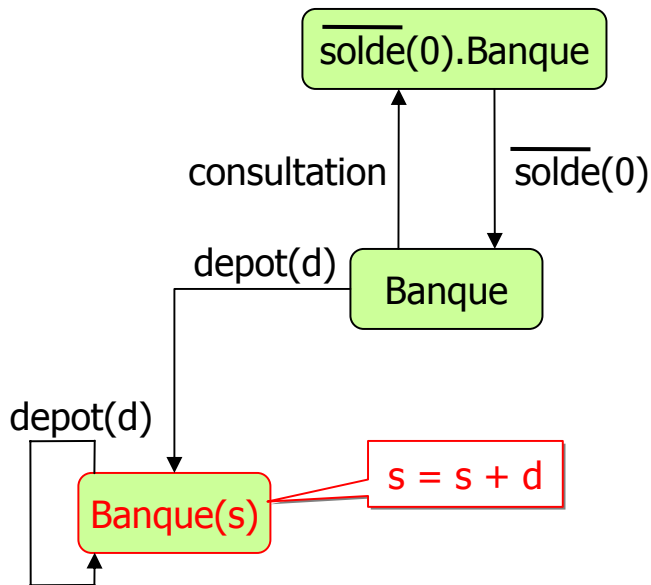


Banque = $\overline{\text{consultation}}.\overline{\text{solde}(0)}.Banque + \text{depot}(d).Banque(d)$

Banque(s) = $\text{depot}(d).Banque(s + d) + \overline{\text{consultation}}.\overline{\text{solde}(s)}.Banque(s) + \text{retrait}(r).Banque(s,r)$

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

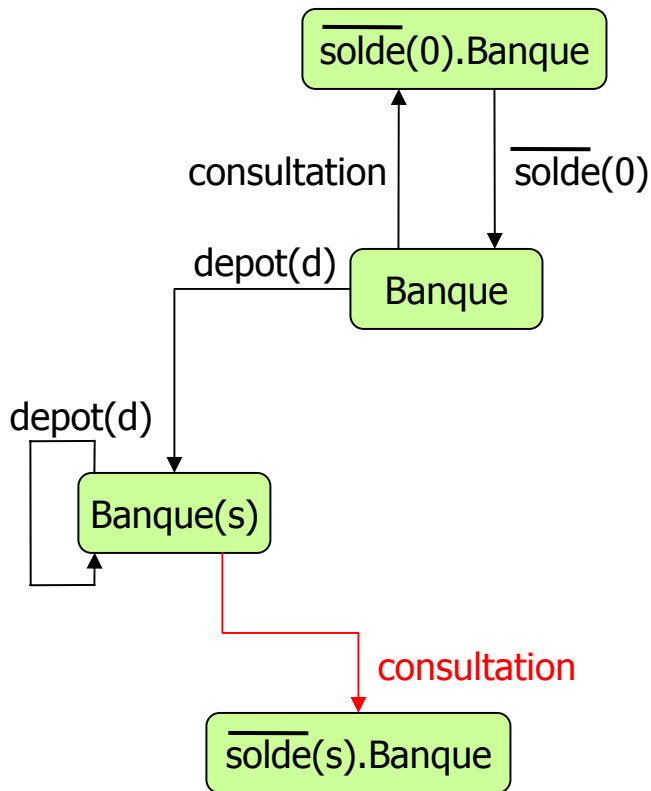


$Banque = \text{consultation}.\overline{\text{solde}(0)}.Banque + \text{depot}(d).Banque(d)$

$Banque(s) = \text{depot}(d).Banque(s + d) + \text{consultation}.\overline{\text{solde}(s)}.Banque(s) + \text{retrait}(r).Banque(s,r)$

$Banque(s,r) = \text{if } (s - r < 0) \text{ then}$
 $\text{erreurRetrait}.Banque(s)$
 $\text{else if } (s - r = 0) \text{ then}$
 $\text{retraitOk}.Banque$
 else
 $\text{retraitOk}.Banque(s - r)$

Pourquoi utiliser le passage de valeurs ?

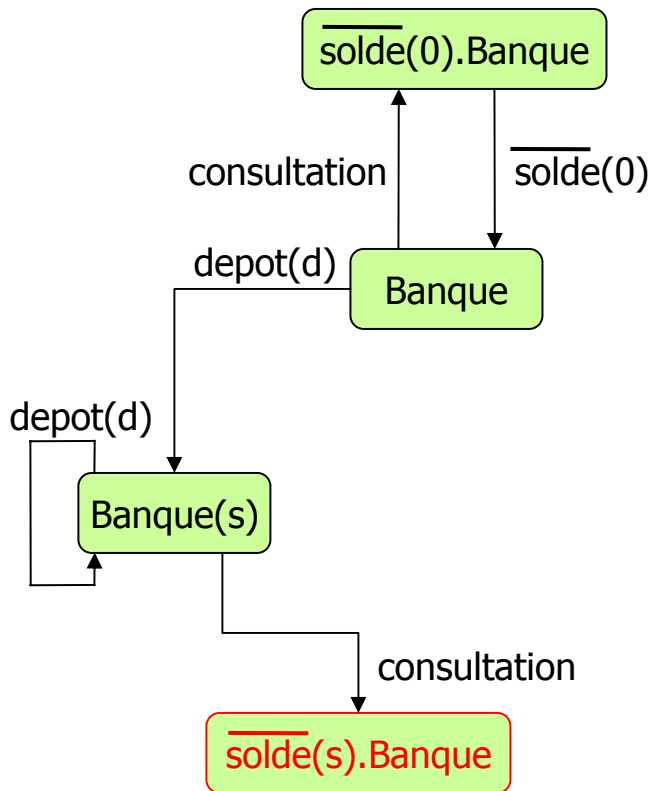


Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde}(s)}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

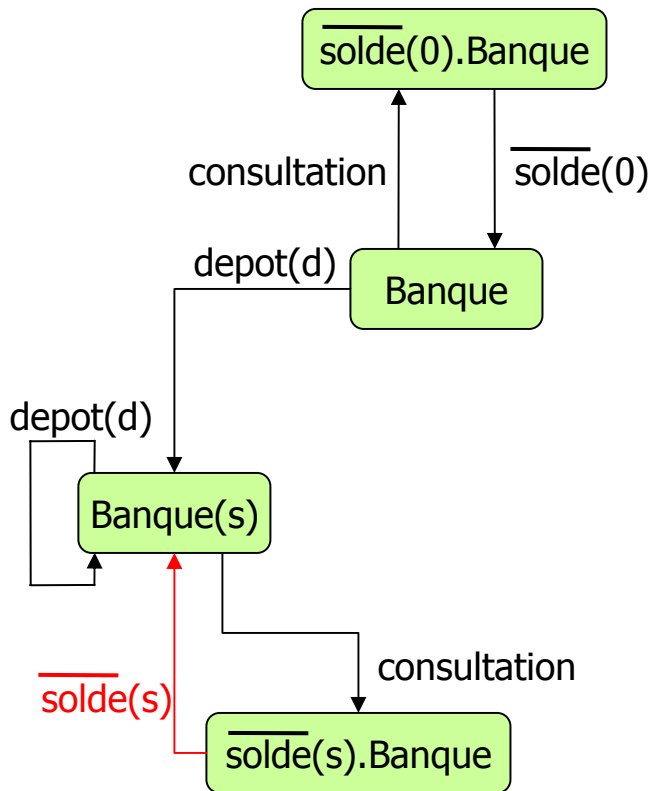


Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde}(s)}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

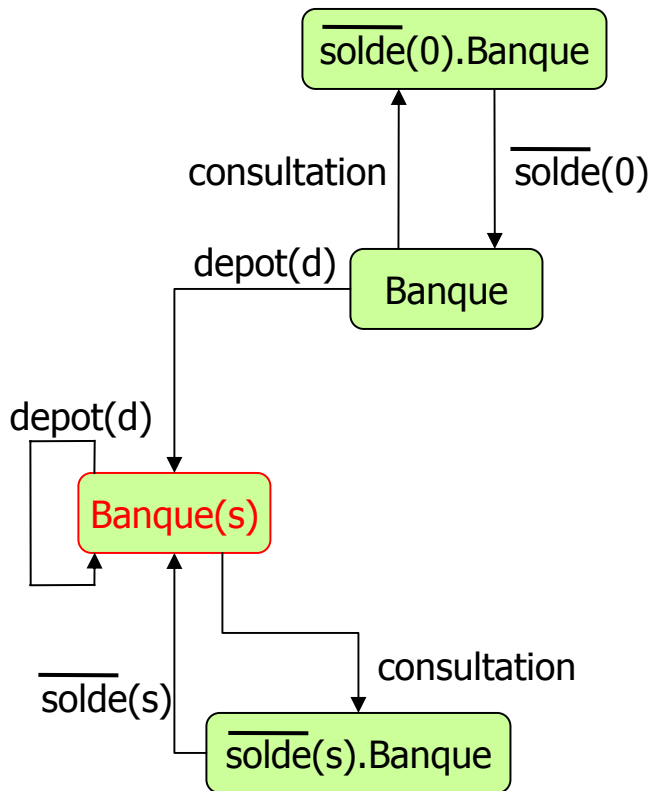


Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde}(s)}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

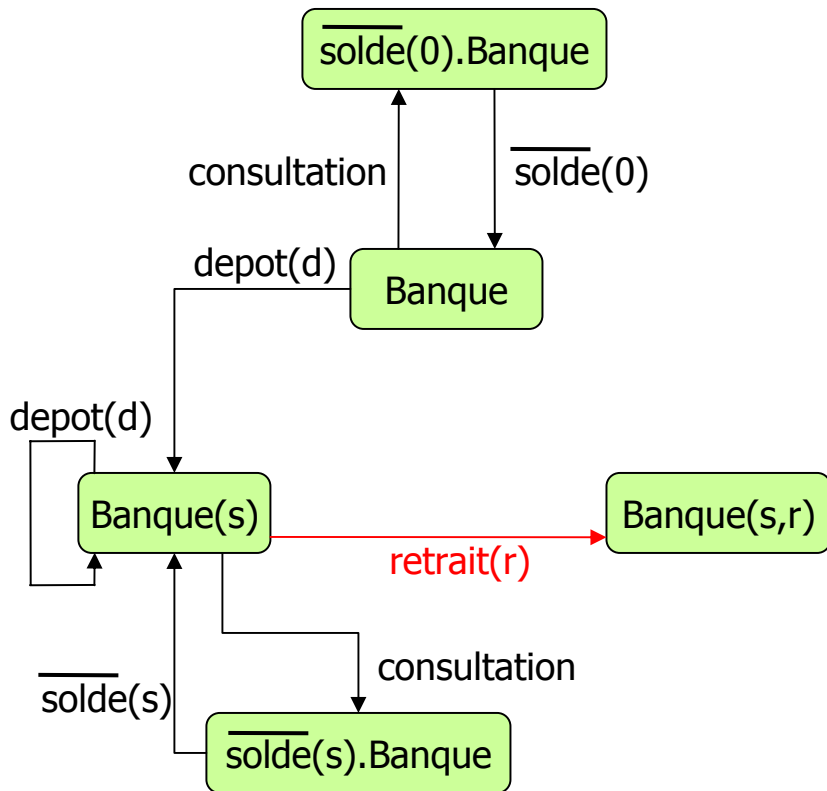


Banque = $\overline{\text{consultation}}.\overline{\text{solde(0)}}.\text{Banque} + \text{depot}(d).\text{Banque}(d)$

Banque(s) = $\text{depot}(d).\text{Banque}(s + d) + \overline{\text{consultation}}.\overline{\text{solde(s)}}.\text{Banque}(s) + \text{retrait}(r).\text{Banque}(s,r)$

Banque(s,r) = if $(s - r < 0)$ then
 $\text{erreurRetrait}.\text{Banque}(s)$
 else if $(s - r = 0)$ then
 $\text{retraitOk}.\text{Banque}$
 else
 $\text{retraitOk}.\text{Banque}(s - r)$

Pourquoi utiliser le passage de valeurs ?

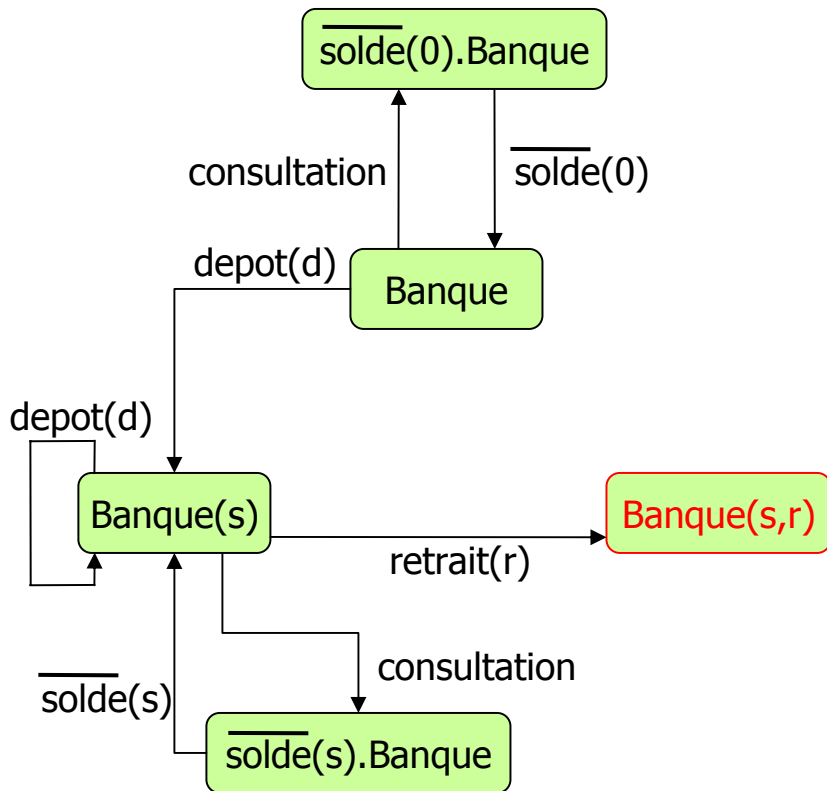


Banque = $\overline{\text{consultation}}.\overline{\text{solde(0)}}.\text{Banque} + \text{depot}(d).\text{Banque}(d)$

Banque(s) = $\text{depot}(d).\text{Banque}(s + d) + \overline{\text{consultation}}.\overline{\text{solde}(s)}.\text{Banque}(s) + \text{retrait}(r).\text{Banque}(s,r)$

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

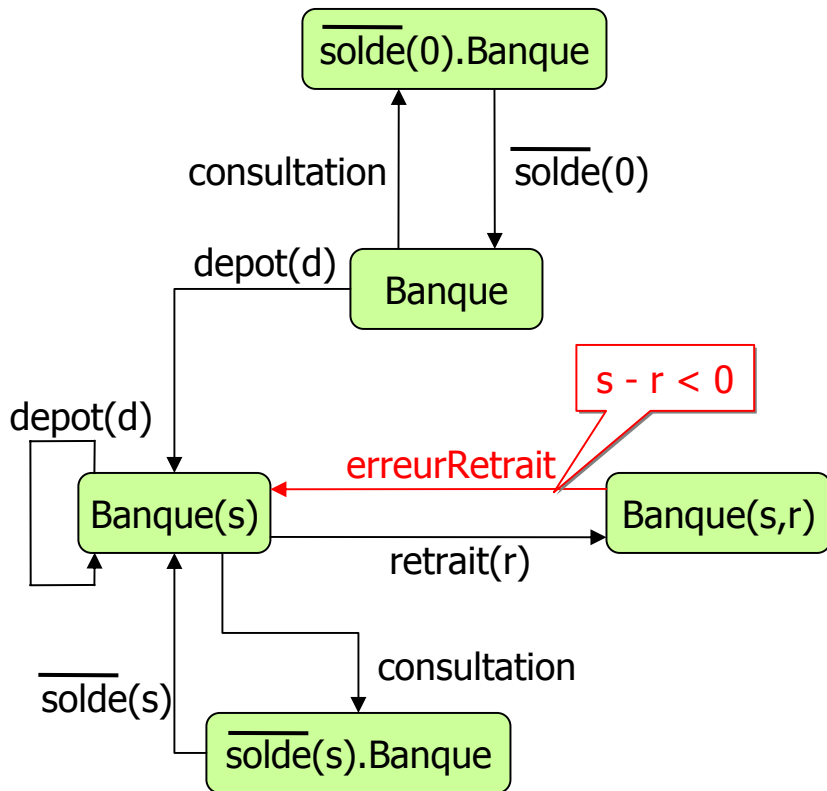


Banque = consultation. $\overline{\text{solde(0)}}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde(s)}}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

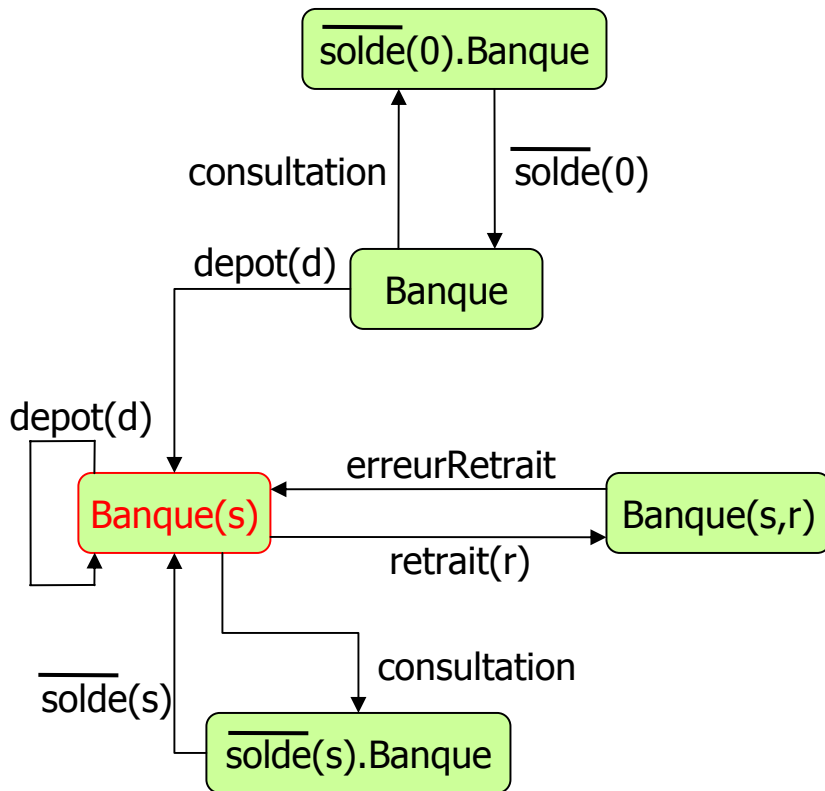


Banque = $\overline{\text{consultation}}.\overline{\text{solde(0)}}.Banque + \text{depot}(d).Banque(d)$

Banque(s) = $\text{depot}(d).Banque(s + d) + \overline{\text{consultation}}.\overline{\text{solde(s)}}.Banque(s) + \text{retrait}(r).Banque(s,r)$

Banque(s,r) = if $(s - r < 0)$ then
 erreurRetrait.Banque(s)
 else if $(s - r = 0)$ then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

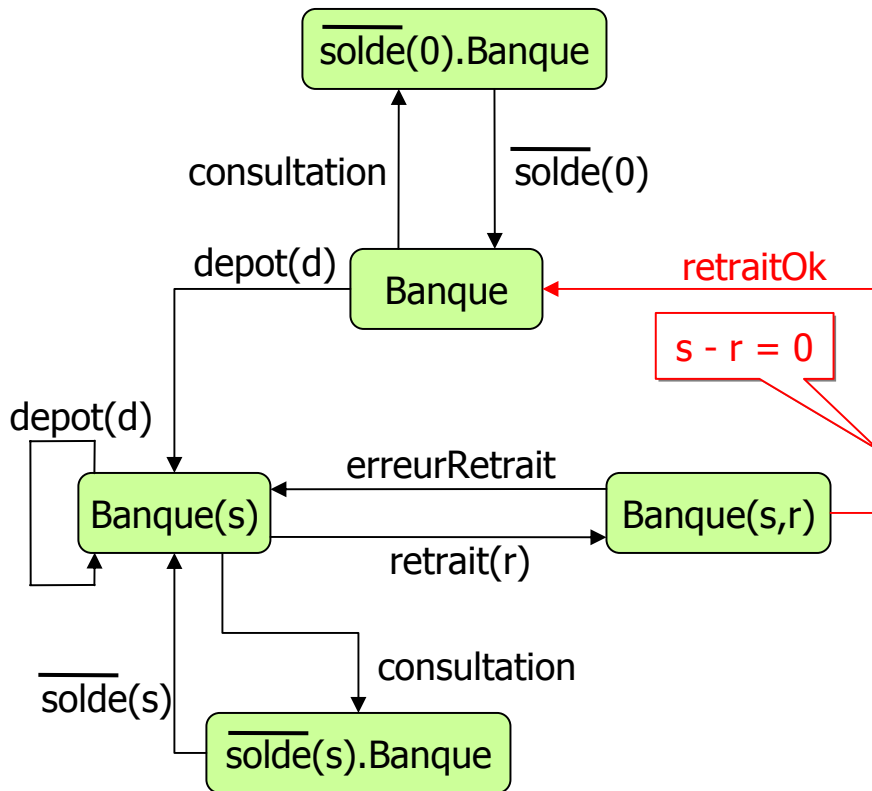


Banque = $\overline{\text{consultation}}.\overline{\text{solde(0)}}.\text{Banque} + \text{depot}(d).\text{Banque}(d)$

Banque(s) = $\text{depot}(d).\text{Banque}(s + d) + \overline{\text{consultation}}.\overline{\text{solde(s)}}.\text{Banque}(s) + \text{retrait}(r).\text{Banque}(s,r)$

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.**Banque(s)**
 else if (s - r = 0) then
 retraitOk.Banque
 else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?

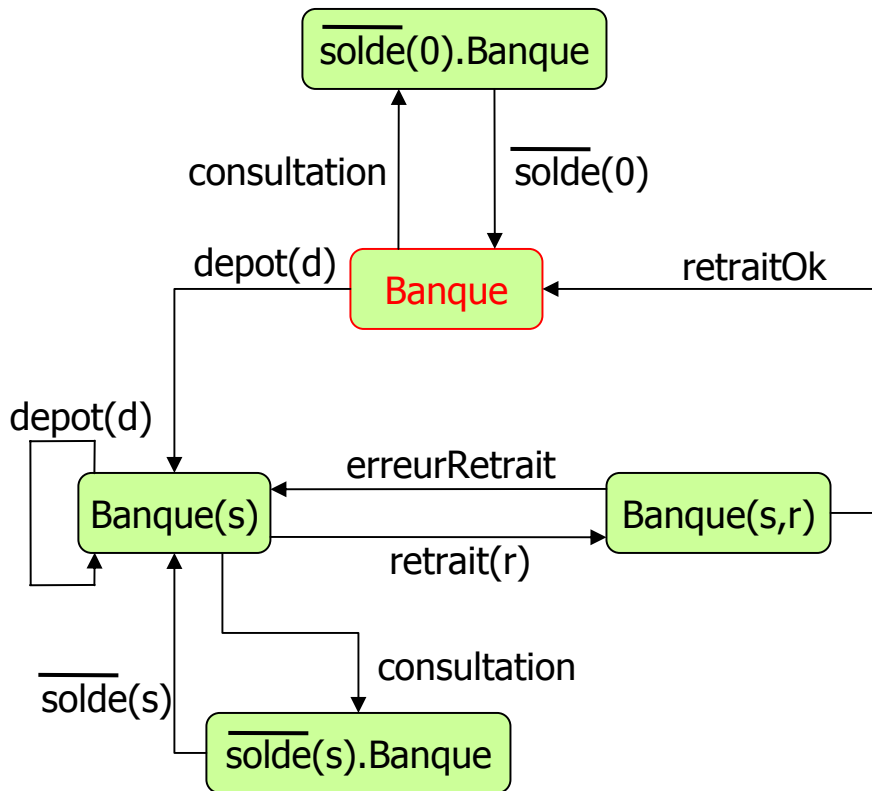


Banque = $\overline{\text{consultation}}.\overline{\text{solde}(0)}.Banque + \text{depot}(d).Banque(d)$

Banque(s) = $\text{depot}(d).Banque(s + d) + \overline{\text{consultation}}.\overline{\text{solde}(s)}.Banque(s) + \text{retrait}(r).Banque(s,r)$

Banque(s,r) = if $(s - r < 0)$ then
 `erreurRetrait.Banque(s)`
 else if $(s - r = 0)$ then
 `retraitOk.Banque`
 else
 `retraitOk.Banque(s - r)`

Pourquoi utiliser le passage de valeurs ?

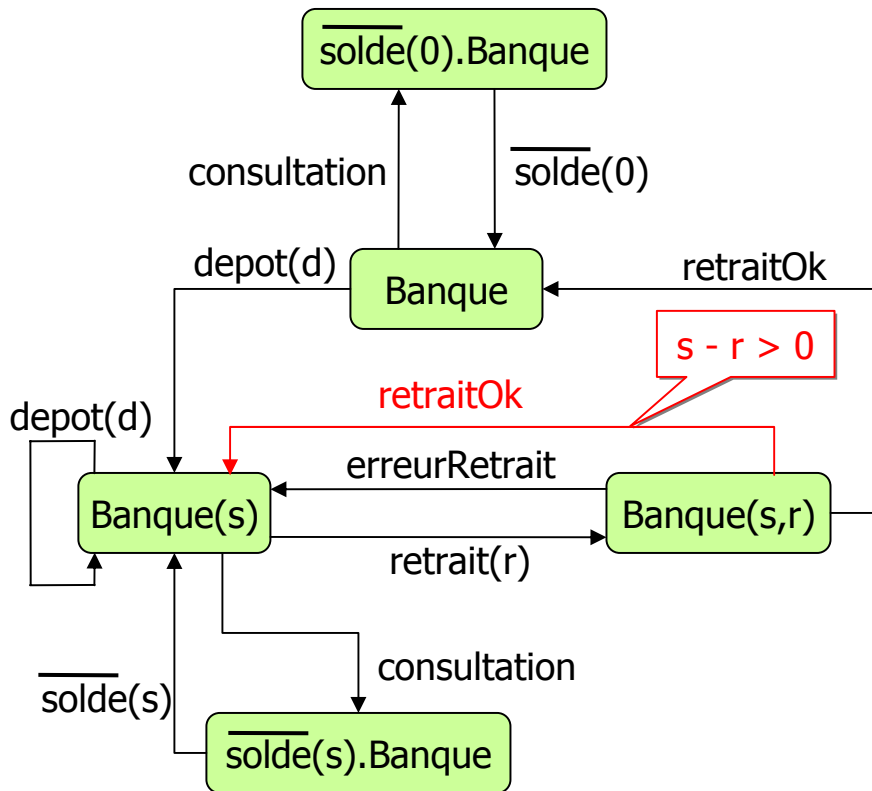


Banque = $\overline{\text{consultation}}.\overline{\text{solde(0)}}.\text{Banque} + \text{depot}(d).\text{Banque}(d)$

$\text{Banque}(s) = \text{depot}(d).\text{Banque}(s + d) + \overline{\text{consultation}}.\overline{\text{solde}(s)}.\text{Banque}(s) + \text{retrait}(r).\text{Banque}(s,r)$

$\text{Banque}(s,r) = \text{if } (s - r < 0) \text{ then } \text{erreurRetrait}.\text{Banque}(s) \text{ else if } (s - r = 0) \text{ then } \text{retraitOk}.\text{Banque} \text{ else } \text{retraitOk}.\text{Banque}(s - r)$

Pourquoi utiliser le passage de valeurs ?

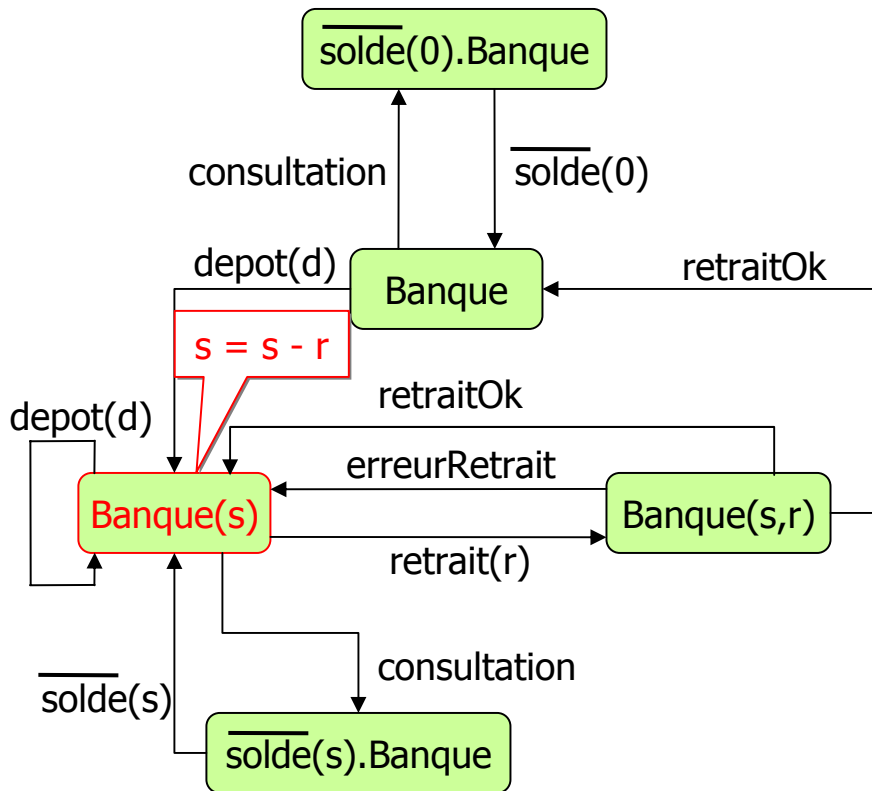


Banque = consultation. $\overline{\text{solde(0)}}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde(s)}}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.Banque(s - r)

Pourquoi utiliser le passage de valeurs ?



Banque = consultation. $\overline{\text{solde}(0)}$.Banque +
depot(d).Banque(d)

Banque(s) = depot(d).Banque(s + d) +
consultation. $\overline{\text{solde}(s)}$.Banque(s) +
retrait(r).Banque(s,r)

Banque(s,r) = if (s - r < 0) then
 erreurRetrait.Banque(s)
else if (s - r = 0) then
 retraitOk.Banque
else
 retraitOk.**Banque(s - r)**

Pourquoi utiliser la notion de coût ?

- Modélisation des critères quantitatifs ou qualitatifs liés à la composition de services en ligne.

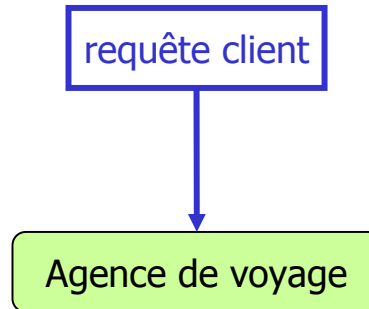
Pourquoi utiliser la notion de coût ?

- Modélisation des critères quantitatifs ou qualitatifs liés à la composition de services en ligne.
- [BCPV04, CCCV05] : Impossibilité de faire une distinction autre que comportementale pour les services en ligne.

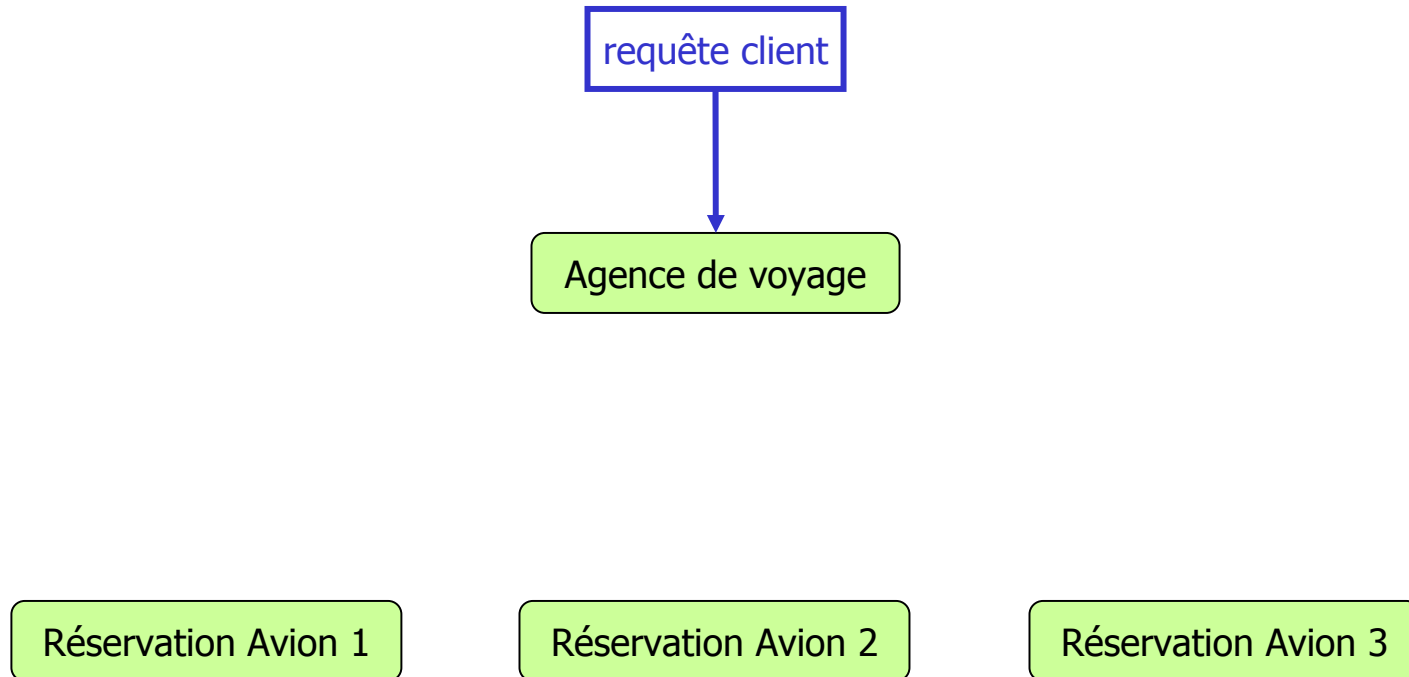
Pourquoi utiliser la notion de coût ?

Agence de voyage

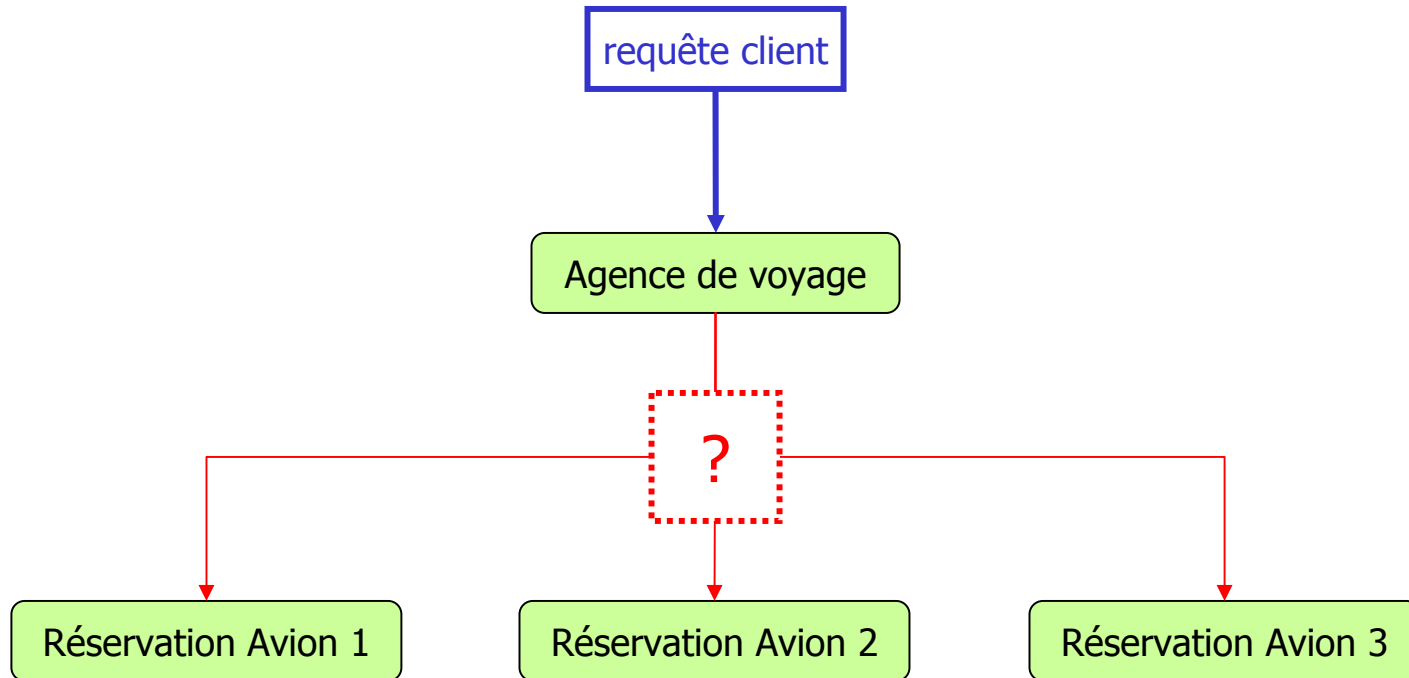
Pourquoi utiliser la notion de coût ?



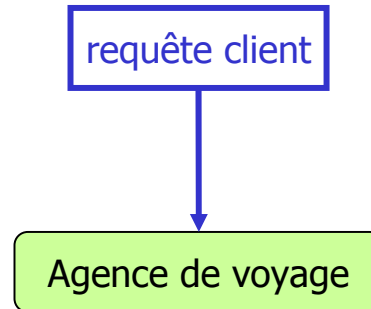
Pourquoi utiliser la notion de coût ?



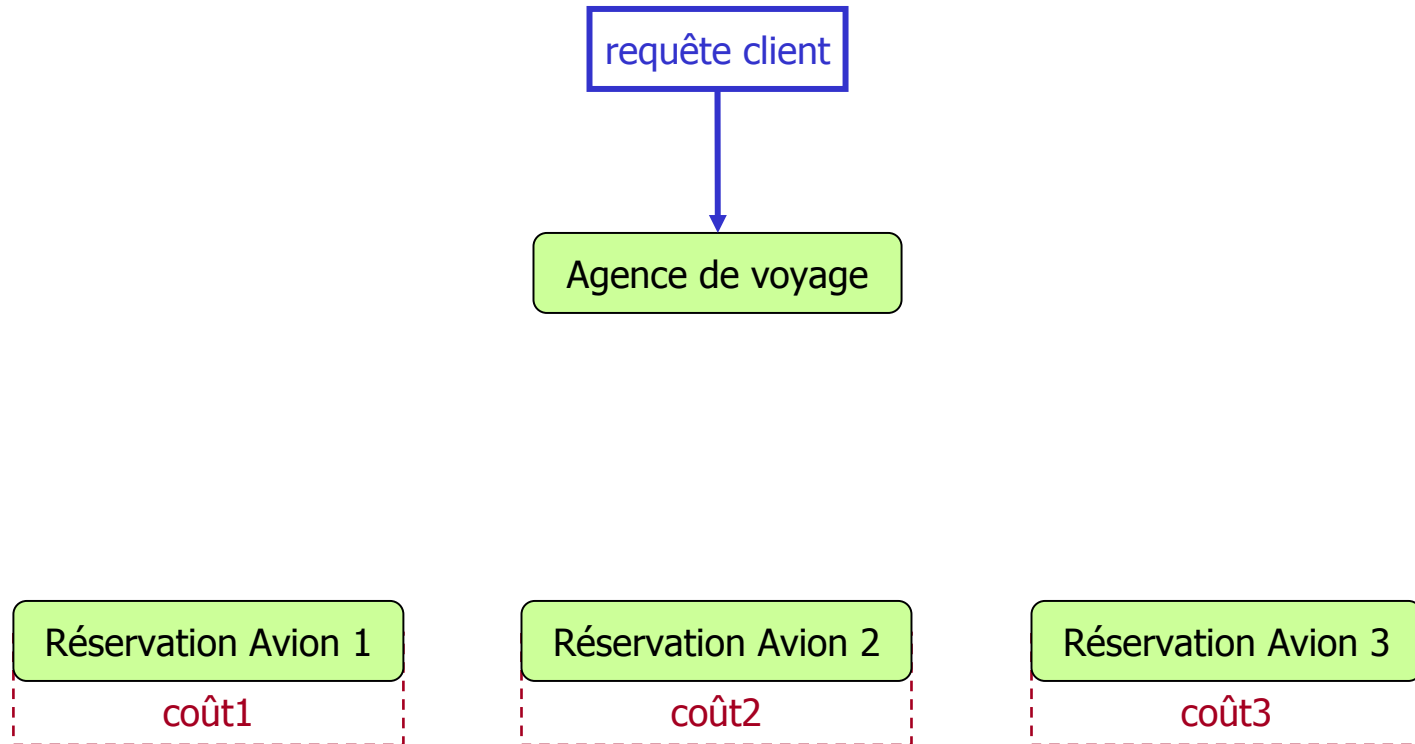
Pourquoi utiliser la notion de coût ?



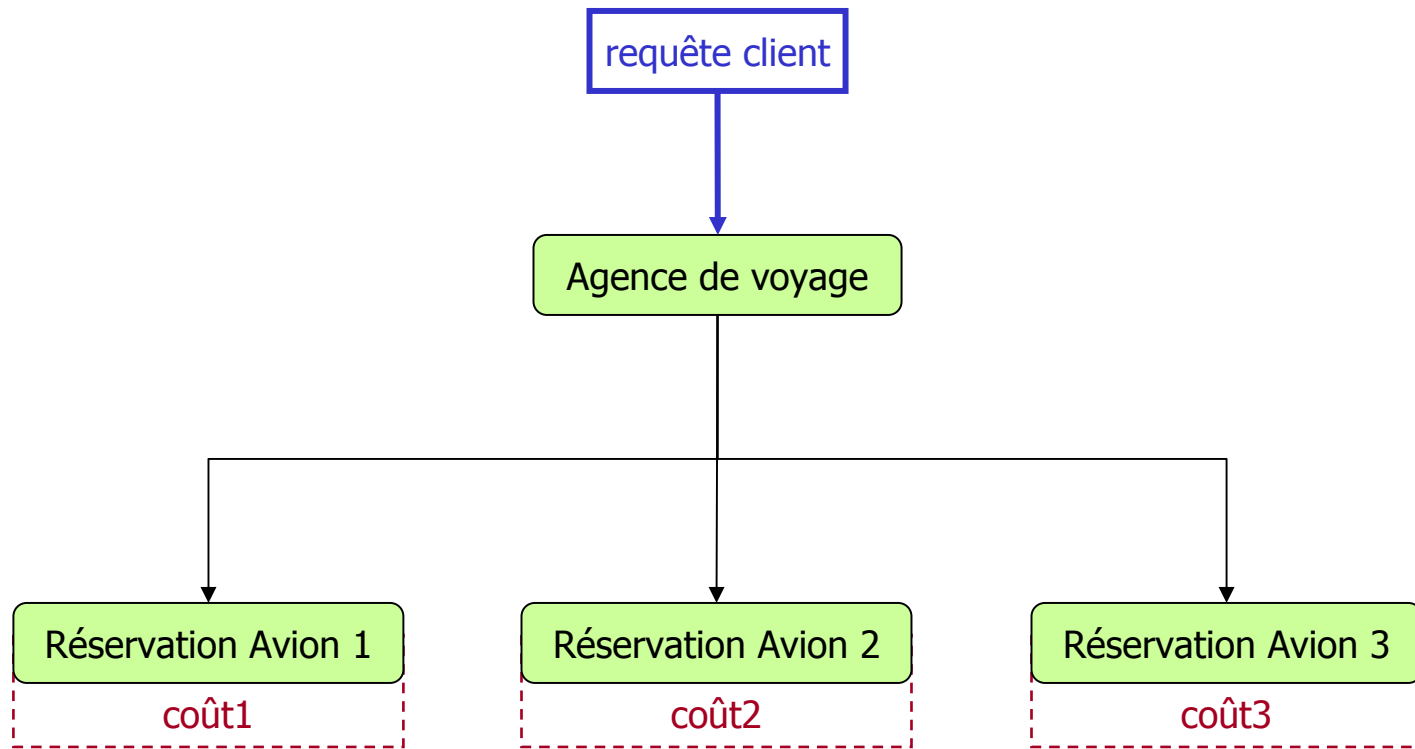
Pourquoi utiliser la notion de coût ?



Pourquoi utiliser la notion de coût ?

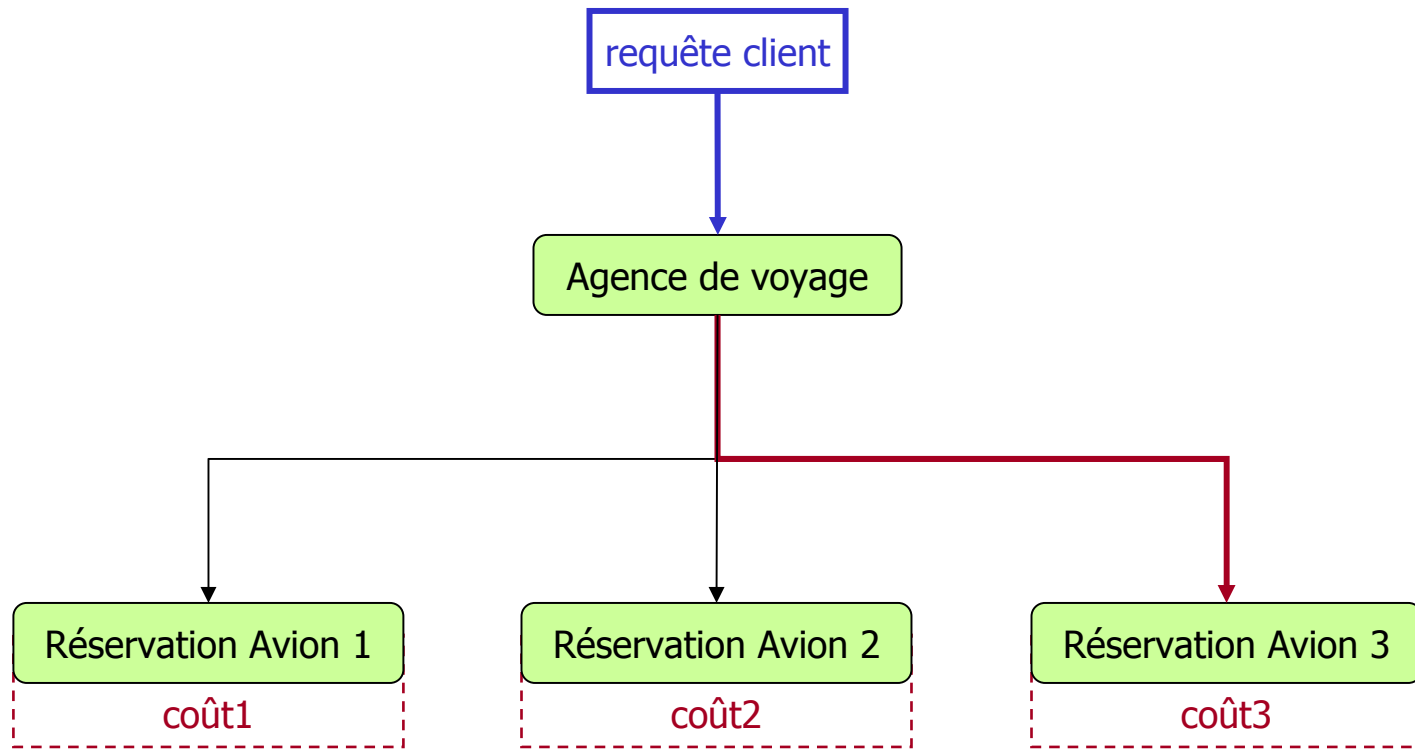


Pourquoi utiliser la notion de coût ?



$$\text{Max}(\text{coût1}, \text{coût2}, \text{coût3}) = \text{coût3}$$

Pourquoi utiliser la notion de coût ?



$$\text{Max}(\text{coût1}, \text{coût2}, \text{coût3}) = \text{coût3}$$

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- **Intégration d'une notion de coût**
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

Intégration d'une notion de coût

- Intégration faite pour l'algèbre PA dans [LS00].
 - Définition d'un ensemble de coûts $\langle C, \oplus, \otimes \rangle$.
 - Intégration de la notion de coût à la relation de transition.
 - Décidabilité de Post*.

Intégration d'une notion de coût

- Intégration faite pour l'algèbre PA dans [LS00].
 - Définition d'un ensemble de coûts $\langle C, \oplus, \otimes \rangle$.
 - Intégration de la notion de coût à la relation de transition.
 - Décidabilité de Post*.
- Modification des règles de la sémantique de CCS :

Intégration d'une notion de coût

- Intégration faite pour l'algèbre PA dans [LS00].
 - Définition d'un ensemble de coûts $\langle C, \oplus, \otimes \rangle$.
 - Intégration de la notion de coût à la relation de transition.
 - Décidabilité de Post*.
- Modification des règles de la sémantique de CCS :

$$\text{Sum}_j \frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j} \quad (j \in I)$$

Intégration d'une notion de coût

- Intégration faite pour l'algèbre PA dans [LS00].
 - Définition d'un ensemble de coûts $\langle C, \oplus, \otimes \rangle$.
 - Intégration de la notion de coût à la relation de transition.
 - Décidabilité de Post*.
- Modification des règles de la sémantique de CCS :

$$\text{Sum}_j \frac{P_j \xrightarrow{\alpha, c_j} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha, c_j} P'_j} \quad (j \in I)$$

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- **Substitutivité de services en ligne**
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

Substitutivité de service en ligne

- Vérification de la substitutivité de services en ligne.

Soient S_1 et S_2 deux services en ligne.

Soient P_1 et P_2 les processus représentant ces deux services.

On dit que S_1 est **substituable** par S_2 si et seulement si :

- $P_1 \approx_s P_2$ (1)

- $\forall \text{tr}^c \in \text{Tr}^c . \text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ (2)

Substitutivité de service en ligne

- Vérification de la substitutivité de services en ligne.

Soient S_1 et S_2 deux services en ligne.
Soient P_1 et P_2 les processus représentant ces deux services.

On dit que S_1 est **substituable** par S_2 si et seulement si :

- $P_1 \approx_s P_2$ (1)

- $\forall \text{tr}^c \in \text{Tr}^c . \text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ (2)

- Décidabilité de (1) et (2) ?

Substitutivité de service en ligne

- [Ber04] Décidabilité de $P_1 \approx_s P_2$ pour CCS avec passage de valeurs.

Substitutivité de service en ligne

- [Ber04] Décidabilité de $P_1 \approx_s P_2$ pour CCS avec passage de valeurs.

– Processus définis par la grammaire suivante :

$$P ::= Q \mid P \setminus L \mid P[f] \mid P|P \mid A(e_1, \dots, e_n)$$
$$Q ::= \sum_{i \in I} \alpha_i.P_i \mid \text{if } b \text{ then } Q \text{ else } Q$$

Substitutivité de service en ligne

- [Ber04] Décidabilité de $P_1 \approx_s P_2$ pour CCS avec passage de valeurs.
 - Processus définis par la grammaire suivante :
$$P ::= Q \mid P \setminus L \mid P[f] \mid P|P \mid A(e_1, \dots, e_n)$$
$$Q ::= \sum_{i \in I} \alpha_i.P_i \mid \text{if } b \text{ then } Q \text{ else } Q$$
 - Utilisation des STGLA pour la définition de la sémantique opérationnelle des processus.

Substitutivité de service en ligne

- [Ber04] Décidabilité de $P_1 \approx_s P_2$ pour CCS avec passage de valeurs.
 - Processus définis par la grammaire suivante :
$$P ::= Q \mid P \setminus L \mid P[f] \mid P|P \mid A(e_1, \dots, e_n)$$
$$Q ::= \sum_{i \in I} \alpha_i.P_i \mid \text{if } b \text{ then } Q \text{ else } Q$$
 - Utilisation des STGLA pour la définition de la sémantique opérationnelle des processus.
 - Adaptation de l'algorithme de vérification de \approx_s proposé dans [LC99].

Substitutivité de service en ligne

- Pour décider de $\text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ pour CCS avec passage de valeurs et notion de coût :

Substitutivité de service en ligne

- Pour décider de $\text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ pour CCS avec passage de valeurs et notion de coût :
 - Intégrer la notion de coût dans la syntaxe et la sémantique de [Ber04].

Substitutivité de service en ligne

- Pour décider de $\text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ pour CCS avec passage de valeurs et notion de coût :
 - Intégrer la notion de coût dans la syntaxe et la sémantique de [Ber04].
 - S'assurer de la représentation des processus sous forme de graphes à branchement fini.

Substitutivité de service en ligne

- Pour décider de $\text{cout}(\text{tr}^c, P_2) \leq f(\text{cout}(\text{tr}^c, P_1))$ pour CCS avec passage de valeurs et notion de coût :
 - Intégrer la notion de coût dans la syntaxe et la sémantique de [Ber04].
 - S'assurer de la représentation des processus sous forme de graphes à branchement fini.
 - Savoir comparer les coûts pour chaque trace complète modulo une fonction f (linéaire).

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- **Compatibilité de services en ligne**
- Compatibilité entre service « client » et composition
- Perspectives

Compatibilité de services en ligne

- Vérification de la compatibilité entre deux services en ligne.

Soient S_1 et S_2 deux services en ligne impliqués dans une composition.

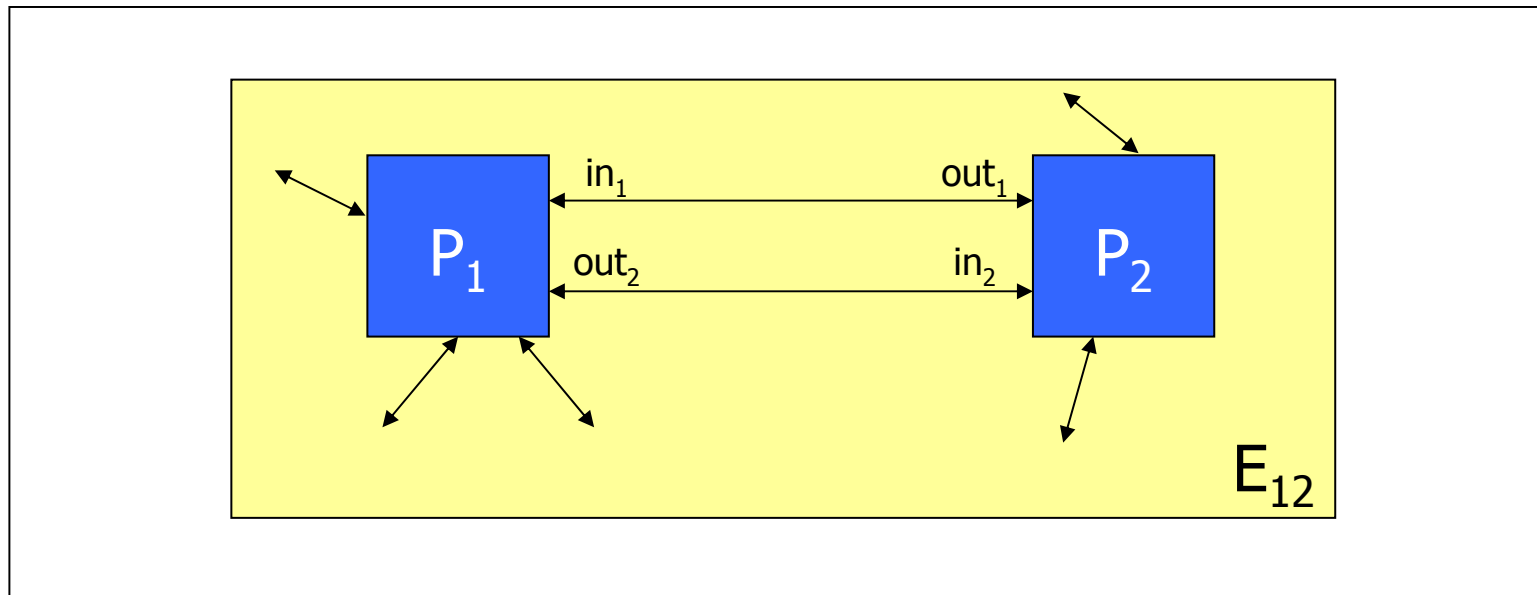
Soient P_1 et P_2 les processus les représentant.

Soit E_{12} un processus représentant l'environnement dans lequel vont évoluer ces services.

On dit que S_1 et S_2 sont **compatibles** si et seulement si la composition $P_1 \mid P_2 \mid E_{12}$ est bien formée.

Compatibilité de services en ligne

- Vérification de la compatibilité entre deux services en ligne.



Compatibilité de services en ligne

- Vérification de la compatibilité entre deux services en ligne.

Soient S_1 et S_2 deux services en ligne impliqués dans une composition.

Soient P_1 et P_2 les processus les représentant.

Soit E_{12} un processus représentant l'environnement dans lequel vont évoluer ces services.

On dit que S_1 et S_2 sont **compatibles** si et seulement si la composition $P_1 \mid P_2 \mid E_{12}$ est bien formée.

Compatibilité de services en ligne

- Processus dit bien formé si et seulement si aucun blocage n'est détecté sur l'ensemble des chemins d'exécution possibles.
- Décidabilité de la bonne formation de $P_1|P_2|E_{12}$?

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:
 - Utilisation de la notion de processus « collé » [FHRR04].

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:
 - Utilisation de la notion de processus « collé » [FHRR04].
 - Redéfinition des règles de la composition parallèle.

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:
 - Utilisation de la notion de processus « collé » [FHRR04].
 - Redéfinition des règles de la composition parallèle.

$$\text{COM}_1 \frac{S_1 \xrightarrow{\alpha} S'_1}{S_1|S_2 \xrightarrow{\alpha} S'_1|S_2}$$

$$\text{COM}_2 \frac{S_2 \xrightarrow{\alpha} S'_2}{S_1|S_2 \xrightarrow{\alpha} S_1|S'_2}$$

$$\text{COM}_3 \frac{S_1 \xrightarrow{\alpha} S'_1 \quad S_2 \xrightarrow{\bar{\alpha}} S'_2}{S_1|S_2 \xrightarrow{\tau} S'_1|S'_2}$$

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:
 - Utilisation de la notion de processus « collé » [FHRR04].
 - Redéfinition des règles de la composition parallèle.

$$\text{COM}_1 \frac{S_1 \xrightarrow{\alpha} S'_1}{S_1|S_2 \xrightarrow{\alpha} S'_1|S_2} (\bullet) \qquad \text{COM}_2 \frac{S_2 \xrightarrow{\alpha} S'_2}{S_1|S_2 \xrightarrow{\alpha} S_1|S'_2} (\bullet)$$

avec $(\bullet) : \alpha, \bar{\alpha} \notin \text{Act}(S_1) \cap \text{Act}(S_2)$

$$\text{COM}_3 \frac{S_1 \xrightarrow{\alpha} S'_1 \quad S_2 \xrightarrow{\bar{\alpha}} S'_2}{S_1|S_2 \xrightarrow{\tau} S'_1|S'_2}$$

Compatibilité de services en ligne

- Pour décider de la bonne formation de $P_1|P_2|E_{12}$:
 - Utilisation de la notion de processus « collé » [FHRR04].
 - Redéfinition des règles de la composition parallèle.

$$\text{COM}_1 \frac{S_1 \xrightarrow{\alpha} S'_1}{S_1|S_2 \xrightarrow{\alpha} S'_1|S_2} (\bullet) \qquad \text{COM}_2 \frac{S_2 \xrightarrow{\alpha} S'_2}{S_1|S_2 \xrightarrow{\alpha} S_1|S'_2} (\bullet)$$

avec $(\bullet) : \alpha, \bar{\alpha} \notin \text{Act}(S_1) \cap \text{Act}(S_2)$

$$\text{COM}_3 \frac{S_1 \xrightarrow{\alpha} S'_1 \quad S_2 \xrightarrow{\bar{\alpha}} S'_2}{S_1|S_2 \xrightarrow{\tau} S'_1|S'_2}$$

- Vérification du non-collage par analyse d'atteignabilité.

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- **Compatibilité entre service « client » et composition**
- Perspectives

Compatibilité entre service « client » et composition

- Vérification de la compatibilité entre un service en ligne et une composition de services.

Soient S un service en ligne et C une composition de services.
Soient P_S et P_C les processus les représentant.

On dit que S et C sont **compatibles** si et seulement si :

- $(P_S)^{-1} \approx_s P_C$ (1)
- $P_S \mid P_C$ bien formé (2)
- $\text{cout}(P_C) \leq \text{limiteClient}$ (ou $\text{cout}(P_C) \geq \text{limiteClient}$) (3)

Compatibilité entre service « client » et composition

- Vérification de la compatibilité entre un service en ligne et une composition de services.

Soient S un service en ligne et C une composition de services.
Soient P_S et P_C les processus les représentant.

On dit que S et C sont **compatibles** si et seulement si :

- $(P_S)^{-1} \approx_s P_C$ (1)
- $P_S \mid P_C$ bien formé (2)
- $\text{cout}(P_C) \leq \text{limiteClient}$ (ou $\text{cout}(P_C) \geq \text{limiteClient}$) (3)

- Décidabilité de (1), (2) et (3) ?

Compatibilité entre service « client » et composition

- [Ber04] Décidabilité de $(P_S)^{-1} \approx_s P_C$ pour CCS avec passage de valeur.

Compatibilité entre service « client » et composition

- [Ber04] Décidabilité de $(P_S)^{-1} \approx_s P_C$ pour CCS avec passage de valeur.
- Décider de la bonne formation de $P_S|P_C$ se ramène à vérifier l'absence de collage.

Compatibilité entre service « client » et composition

- [Ber04] Décidabilité de $(P_S)^{-1} \approx_s P_C$ pour CCS avec passage de valeur.
- Décider de la bonne formation de $P_S|P_C$ se ramène à vérifier l'absence de collage.
- Décider de $\text{cout}(P_C) \leq \text{limiteClient}$ passe par la décidabilité de la recherche du coût maximal sur l'ensemble des traces complètes d'un processus.

Plan de la présentation

- Motivations
- Algèbres de processus et services en ligne
- Intégration d'une notion de coût
- Substitutivité de services en ligne
- Compatibilité de services en ligne
- Compatibilité entre service « client » et composition
- Perspectives

Perspectives

- Substitutivité de services en ligne :
 - Adaptation de la notion de STGLA pour tenir compte des coûts.
 - Utilisation des automates avec poids pour la comparaison des coûts des traces complètes.

Perspectives

- Substitutivité de services en ligne :
 - Adaptation de la notion de STGLA pour tenir compte des coûts.
 - Utilisation des automates avec poids pour la comparaison des coûts des traces complètes.
- Compatibilité entre services en ligne :
 - Utilisation de la notion de processus « collé » pour une vérification par analyse d'atteignabilité.

Perspectives

- Compatibilité « client » / composition :
 - Recherche d'un coût maximal sur un ensemble de trace complète.

Perspectives

- Compatibilité « client » / composition :
 - Recherche d'un coût maximal sur un ensemble de trace complète.
- Implémentation (extension de CWB-NC).

Perspectives

- Etudier les dépendances éventuelles entre valeurs et coûts.

Perspectives

- Etudier les dépendances éventuelles entre valeurs et coûts.

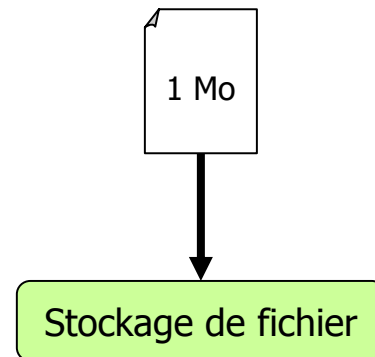
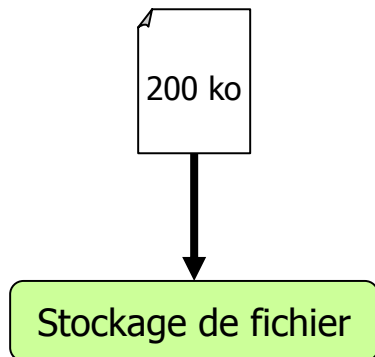


200 ko

1 Mo

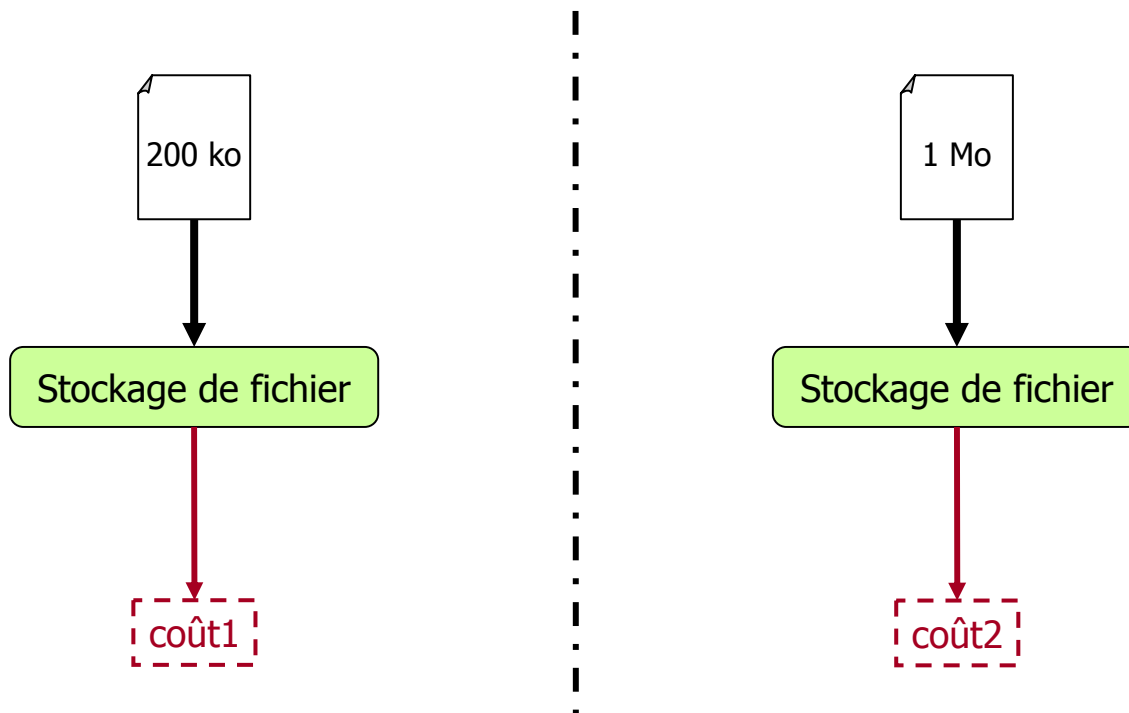
Perspectives

- Etudier les dépendances éventuelles entre valeurs et coûts.



Perspectives

- Etudier les dépendances éventuelles entre valeurs et coûts.



Perspectives

- Adaptation de cette approche à CryptoSPA.
- Vérification de la sécurité des échanges d'informations entre services en ligne.