

# Intégration de la qualité de service dans la vérification de la substitutivité des services Web

Jérôme Voinot

Laboratoire d'Informatique de l'Université de Franche-Comté

ARA COPS

29/30 mars 2007 – Nancy

# Plan de la présentation

Motivations

Modélisation des services Web et des aspects de qualité de service

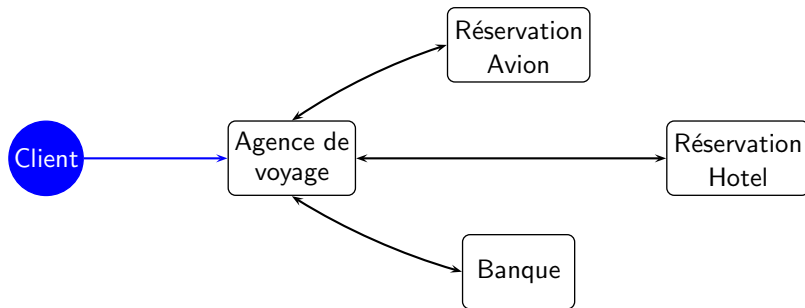
Vérification de la substitutivité des services Web

Implémentation de l'approche

Perspectives

# Motivations

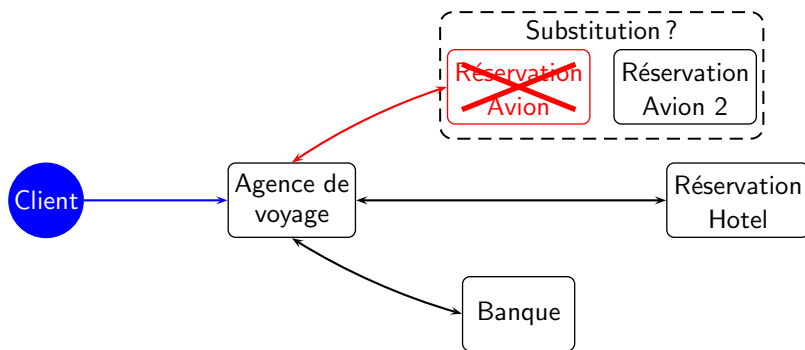
Utilisation des services Web en tant que composants d'applications.



Implique la possibilité de pouvoir remplacer à tout moment un service par un autre (défaillance, ...).

# Motivations

Utilisation des services Web en tant que composants d'applications.



Implique la possibilité de pouvoir remplacer à tout moment un service par un autre (défaillance, ...).

Tenir compte de critères qualitatifs et/ou quantitatifs dans le cadre de la substitution des services Web (audit de services pour leurs définitions).

→ *Ne pas substituer un service par un autre si ces deux services n'ont pas des performances équivalentes.*

Rester proche des normes qui ont cours dans le monde des services Web.

Tenir compte de critères qualitatifs et/ou quantitatifs dans le cadre de la substitution des services Web (audit de services pour leurs définitions).

→ *Ne pas substituer un service par un autre si ces deux services n'ont pas des performances équivalentes.*

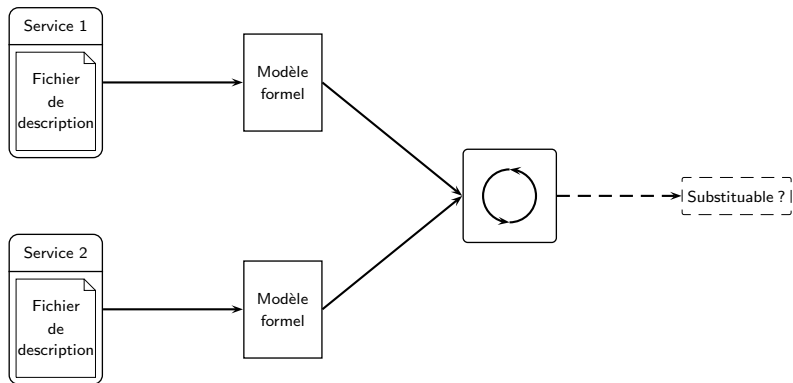
Rester proche des normes qui ont cours dans le monde des services Web.

Tenir compte de critères qualitatifs et/ou quantitatifs dans le cadre de la substitution des services Web (audit de services pour leurs définitions).

→ *Ne pas substituer un service par un autre si ces deux services n'ont pas des performances équivalentes.*

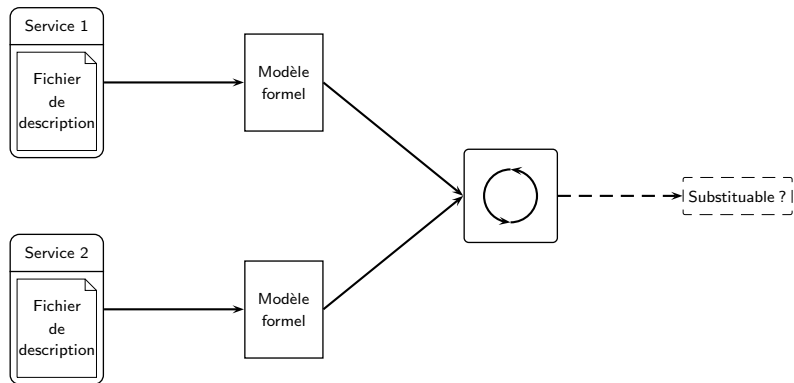
Rester proche des normes qui ont cours dans le monde des services Web.

# Schéma de vérification



Comment tenir compte de la qualité de service avec les normes actuelles ?  
Quel modèle formel pour représenter les services Web ?  
Comment dire que deux services Web sont substituables ?

# Schéma de vérification

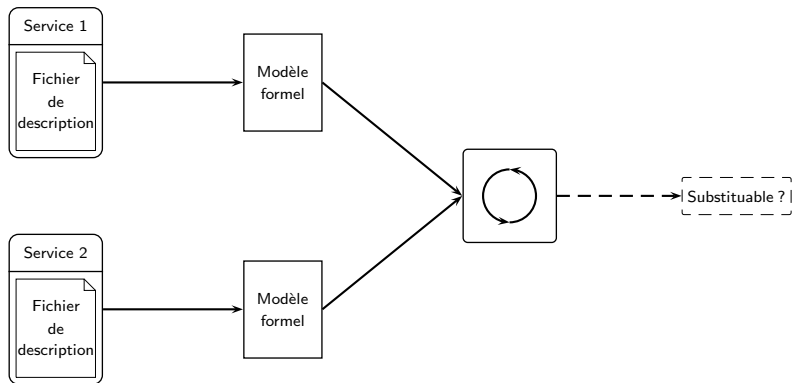


Comment tenir compte de la qualité de service avec les normes actuelles ?

Quel modèle formel pour représenter les services Web ?

Comment dire que deux services Web sont substituables ?

# Schéma de vérification

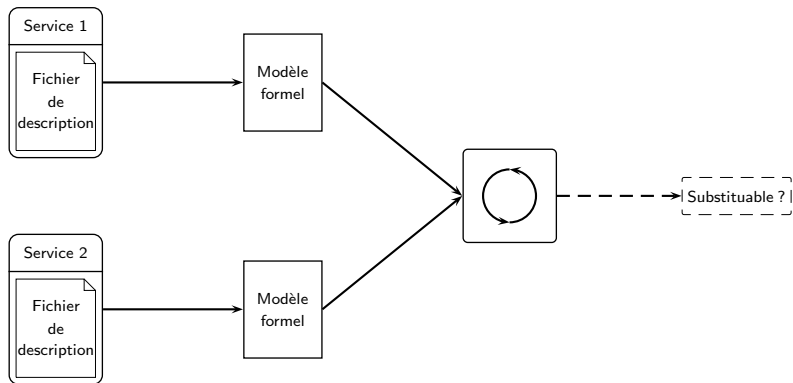


Comment tenir compte de la qualité de service avec les normes actuelles ?

Quel modèle formel pour représenter les services Web ?

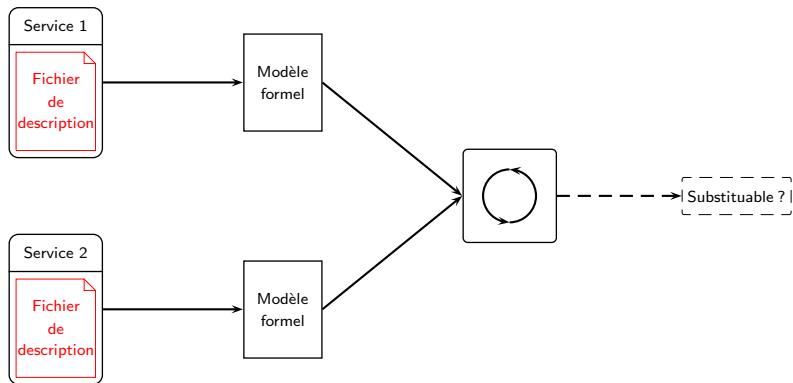
Comment dire que deux services Web sont substituables ?

# Schéma de vérification



Comment tenir compte de la qualité de service avec les normes actuelles ?  
Quel modèle formel pour représenter les services Web ?  
Comment dire que deux services Web sont substituables ?

# Schéma de vérification



Comment tenir compte de la qualité de service avec les normes actuelles ?

Quel modèle formel pour représenter les services Web ?

Comment dire que deux services Web sont substituables ?

Définition de ce qu'est la qualité de service dans le cadre des services Web par W3C et OASIS.

Deux optiques de mise en place de la qualité de service :

- Définition de nouveaux langages de spécification :
  - WSLA [KL02],
  - WSML [SDM02],
  - ...
- Extension des normes actuelles :
  - WS-QoS [Tia05],
  - UDDIe [SRAAW03],
  - ...

Définition de ce qu'est la qualité de service dans le cadre des services Web par W3C et OASIS.

Deux optiques de mise en place de la qualité de service :

- Définition de nouveaux langages de spécification :
  - WSLA [KL02],
  - WSML [SDM02],
  - ...
- Extension des normes actuelles :
  - WS-QoS [Tia05],
  - UDDIe [SRAAW03],
  - ...

Définition de ce qu'est la qualité de service dans le cadre des services Web par W3C et OASIS.

Deux optiques de mise en place de la qualité de service :

- Définition de nouveaux langages de spécification :
  - WSLA [KL02],
  - WSML [SDM02],
  - ...
- Extension des normes actuelles :
  - WS-QoS [Tia05],
  - UDDIe [SRAAW03],
  - ...

Définition de ce qu'est la qualité de service dans le cadre des services Web par W3C et OASIS.

Deux optiques de mise en place de la qualité de service :

- Définition de nouveaux langages de spécification :
  - WSLA [KL02],
  - WSML [SDM02],
  - ...
- Extension des normes actuelles :
  - WS-QoS [Tia05],
  - UDDIe [SRAAW03],
  - ...

# Extension de BPEL4WS 1.1

Introduction d'aspects de qualité de service dans BPEL (inspiré de [BPB06]).

Extension des activités invoke, receive, reply et onMessage avec un nouvel élément :

```
<costs>  
  <cost type="executionTime" value="..." />  
  <cost type="financialCost" value="..." />  
</costs>
```

# Extension de BPEL4WS 1.1

Introduction d'aspects de qualité de service dans BPEL (inspiré de [BPB06]).

Extension des activités invoke, receive, reply et onMessage avec un nouvel élément :

```
<costs>  
  <cost type="executionTime" value="..." />  
  <cost type="financialCost" value="..." />  
</costs>
```

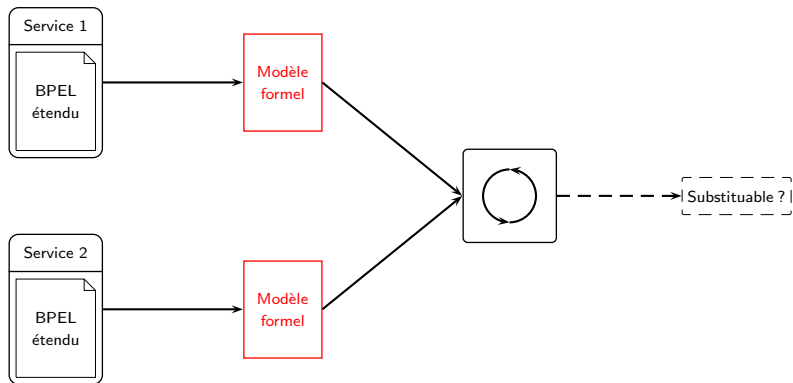
# Extension de BPEL4WS 1.1

Introduction d'aspects de qualité de service dans BPEL (inspiré de [BPB06]).

Extension des activités invoke, receive, reply et onMessage avec un nouvel élément :

```
<costs>  
  <cost type="executionTime" value="..." />  
  <cost type="financialCost" value="..." />  
</costs>
```

# Schéma de vérification



Comment tenir compte de la qualité de service avec les normes actuelles ?

Quel modèle formel pour représenter les services Web ?

Comment dire que deux services Web sont substituables ?

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

Utilisation de différents modèles formels tels que :

- Réseaux de Petri [HB03]
- Algèbres de processus [CS06]
- Automates finis [Fos06]

Modélisation basée sur une extension des automates finis car :

- Normes décrites sous formes de schémas XSD.
- Schéma XSD décrit un langage régulier.

# Automate avec coût de service

Utilisation des automates max-plus permettant de tenir compte des spécificités de BPEL étendu.

## Définition (Automate avec coût de service)

*Etant donné un ensemble non-vide  $C \in \mathbb{Q}$  de coûts de service, un automate avec coût de service sur  $\Sigma$  est un sextuple*

*$\mathcal{A} = (S, \Sigma, \delta, s_0, s_f, A)$ , où :*

- *$S$  est un ensemble fini non-vide d'états,*
- *$\Sigma$  est un ensemble fini non-vide d'actions,*
- *$\delta \subseteq S \times \Sigma \times C \times S$  est la relation de transition,*
- *$s_0 \in S$  est l'état initial,*
- *$s_f \in S$  est l'état final, et*
- *$A \subseteq S$  est un ensemble fini (éventuellement vide) d'états d'acceptation intermédiaire.*

# Automate avec coût de service

Utilisation des automates max-plus permettant de tenir compte des spécificités de BPEL étendu.

## Définition (Automate avec coût de service)

*Etant donné un ensemble non-vide  $C \in \mathbb{Q}$  de coûts de service, un automate avec coût de service sur  $\Sigma$  est un sextuple*

*$A = (S, \Sigma, \delta, s_0, s_f, A)$ , où :*

- *$S$  est un ensemble fini non-vide d'états,*
- *$\Sigma$  est un ensemble fini non-vide d'actions,*
- *$\delta \subseteq S \times \Sigma \times C \times S$  est la relation de transition,*
- *$s_0 \in S$  est l'état initial,*
- *$s_f \in S$  est l'état final, et*
- *$A \subseteq S$  est un ensemble fini (éventuellement vide) d'états d'acceptation intermédiaire.*

## Définition (Chemin réussi)

Etant donné un automate avec coût de service  $\mathcal{A}$ , un chemin réussi dans  $\mathcal{A}$  est une séquence  $\pi = t_0, t_1, \dots, t_n$  où  $t_i = (q_i, a_i, c_i, q_{i+1}) \in \delta$ ,  $q_0 = s_0$  et  $q_n = s_f$  ou  $q_n \in A$ . On note  $\Pi = \{\pi_1, \dots, \pi_n, \dots\}$  l'ensemble des chemins réussi dans  $\mathcal{A}$ .

La trace d'un chemin réussi est définie comme étant :

$$\text{trace}(\pi) = a_0.a_1.\dots.a_n$$

et le coût de  $\pi$  dans  $\mathcal{A}$  comme étant :

$$\text{cost}_{\mathcal{A}}(\pi) = \sum_{i=1}^n c_i$$

# Coût d'un mot dans un automate avec coût de service

Le langage reconnu par  $\mathcal{A}$ , dénoté  $L(\mathcal{A})$ , est l'ensemble des traces des chemins réussis dans  $\mathcal{A}$ . Chaque élément  $u \in L(\mathcal{A})$  est appelé un mot.

## Définition (Coût d'un mot)

*Etant donné un automate avec coût de service  $\mathcal{A}$ , le coût d'un mot  $u \in L(\mathcal{A})$  est donné par la fonction :*

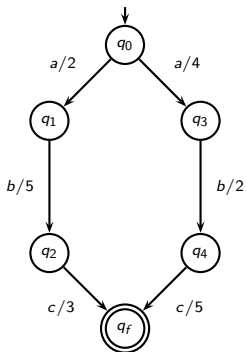
$$\text{cost}_{\mathcal{A}}: L(\mathcal{A}) \rightarrow \mathbb{Q}$$

*définie par :*

$$\max\{\text{cost}_{\mathcal{A}}(\pi) \mid \pi \in \Pi \wedge \text{trace}(\pi) = u\}$$

# Exemple du coût d'un mot

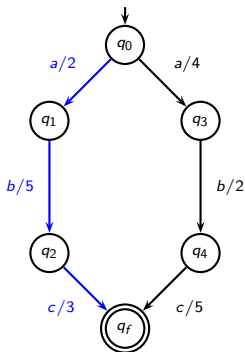
Coût du mot  $u = a.b.c$  ?



# Exemple du coût d'un mot

Coût du mot  $u = a.b.c$  ?

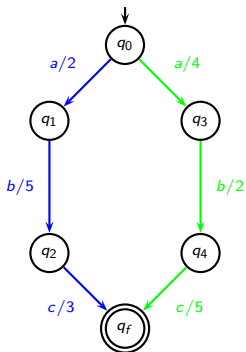
$$\text{cost}_{\mathcal{A}}(\pi_1) = 10$$



# Exemple du coût d'un mot

Coût du mot  $u = a.b.c$  ?

$$\text{cost}_{\mathcal{A}}(\pi_1) = 10$$

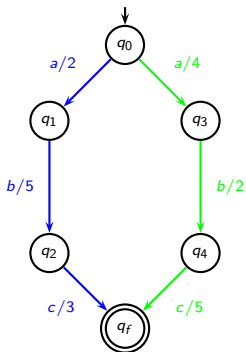


$$\text{cost}_{\mathcal{A}}(\pi_2) = 11$$

# Exemple du coût d'un mot

Coût du mot  $u = a.b.c$  ?

$$\text{cost}_{\mathcal{A}}(\pi_1) = 10$$



$$\text{cost}_{\mathcal{A}}(\pi_2) = 11$$

$$\text{cost}_{\mathcal{A}}(u) = \max\{\text{cost}_{\mathcal{A}}(\pi_1), \text{cost}_{\mathcal{A}}(\pi_2)\}$$

## Définition de règles de traductions permettant de définir :

- Actions de l'automate :
  - Actions de communication, notées  $op:pt:p$ .
  - Actions silencieuses, notées  $\tau$ .
  - Actions de sortie de boucle, notées  $\surd$ .
  - Actions de terminaison, notées  $\ominus$ .
- Transitions de l'automate :
  - Transitions d'envois.
  - Transitions de réceptions.
  - Tau-transitions.
  - Transitions de sortie de boucle.
  - Transitions de terminaison.
- Enchaînement des transitions de l'automate.

Définition de règles de traductions permettant de définir :

- Actions de l'automate :
  - Actions de communication, notées  $op:pt:p$ .
  - Actions silencieuses, notées  $\tau$ .
  - Actions de sortie de boucle, notées  $\surd$ .
  - Actions de terminaison, notées  $\ominus$ .
- Transitions de l'automate :
  - Transitions d'envois.
  - Transitions de réceptions.
  - Tau-transitions.
  - Transitions de sortie de boucle.
  - Transitions de terminaison.
- Enchaînement des transitions de l'automate.

Définition de règles de traductions permettant de définir :

- Actions de l'automate :
  - Actions de communication, notées  $op:pt:p$ .
  - Actions silencieuses, notées  $\tau$ .
  - Actions de sortie de boucle, notées  $\surd$ .
  - Actions de terminaison, notées  $\ominus$ .
- Transitions de l'automate :
  - Transitions d'envois.
  - Transitions de réceptions.
  - Tau-transitions.
  - Transitions de sortie de boucle.
  - Transitions de terminaison.
- Enchaînement des transitions de l'automate.

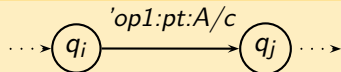
Définition de règles de traductions permettant de définir :

- Actions de l'automate :
  - Actions de communication, notées  $op:pt:p$ .
  - Actions silencieuses, notées  $\tau$ .
  - Actions de sortie de boucle, notées  $\surd$ .
  - Actions de terminaison, notées  $\ominus$ .
- Transitions de l'automate :
  - Transitions d'envois.
  - Transitions de réceptions.
  - Tau-transitions.
  - Transitions de sortie de boucle.
  - Transitions de terminaison.
- Enchaînement des transitions de l'automate.

## Activité BPEL

```
<invoke partnerLink="A" portType="pt" operation="op1" inputVariable="x">  
  <costs>  
    <cost type="executionTime" value="3" />  
    <cost type="financialCost" value="25" />  
  </costs>  
</invoke>
```

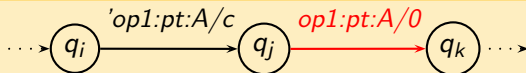
## Automate



## Activité BPEL

```
<invoke partnerLink="A" portType="pt" operation="op1" inputVariable="x"
  outputVariable="y" >
  <costs>
    <cost type="executionTime" value="3" />
    <cost type="financialCost" value="25" />
  </costs>
</invoke>
```

## Automate

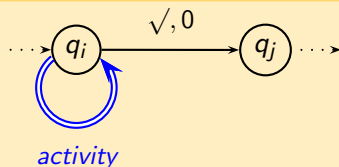


# De BPEL aux automates avec coût de service

## Activité BPEL

```
<while condition=" booleanCondition" >  
  activity  
</while>
```

## Automate

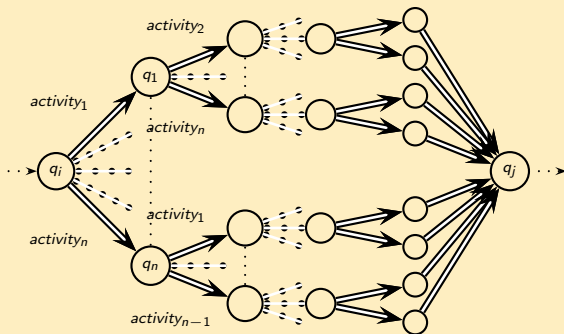


# De BPEL aux automates avec coût de service

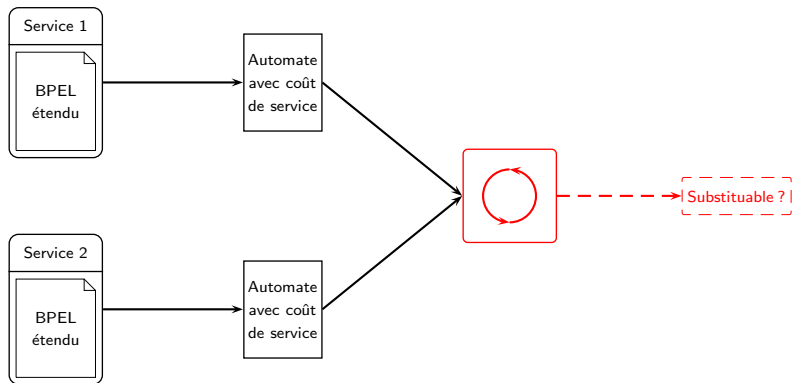
## Activité BPEL

```
<flow>  
  activity1  
  ...  
  activityn  
</flow>
```

## Automate



# Schéma de vérification



Comment tenir compte de la qualité de service avec les normes actuelles ?

Quel modèle formel pour représenter les services Web ?

Comment dire que deux services Web sont substituables ?

# (a,b)-substitutivité

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre si :

- le remplaçant fournit au moins les mêmes services que le service Web à remplacer, et
- les coûts des deux services Web sont similaires considérant deux valeurs  $a, b \in \mathbb{Q}$ .

## Définition ((a, b)-substitutivité)

Soient  $a, b \in \mathbb{Q}$ . Etant donné deux automates avec coût de service  $\mathcal{A}_1$  et  $\mathcal{A}_2$  représentant respectivement deux services Web  $S_1$  et  $S_2$ , on dit que  $S_1$  peut être (a, b)-substitué par  $S_2$  si :

- 1  $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ , et
- 2  $\forall u \in L(\mathcal{A}_1), \text{cost}_{\mathcal{A}_2}(u) \leq a \cdot \text{cost}_{\mathcal{A}_1}(u) + b$

# (a,b)-substitutivité

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre si :

- le remplaçant fournit au moins les mêmes services que le service Web à remplacer, et
- les coûts des deux services Web sont similaires considérant deux valeurs  $a, b \in \mathbb{Q}$ .

## Définition ((a, b)-substitutivité)

*Soient  $a, b \in \mathbb{Q}$ . Etant donné deux automates avec coût de service  $\mathcal{A}_1$  et  $\mathcal{A}_2$  représentant respectivement deux services Web  $S_1$  et  $S_2$ , on dit que  $S_1$  peut être (a, b)-substitué par  $S_2$  si :*

- 1  $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ , et
- 2  $\forall u \in L(\mathcal{A}_1), \text{cost}_{\mathcal{A}_2}(u) \leq a \cdot \text{cost}_{\mathcal{A}_1}(u) + b$

# (a,b)-substitutivité

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre si :

- le remplaçant fournit au moins les mêmes services que le service Web à remplacer, et
- les coûts des deux services Web sont similaires considérant deux valeurs  $a, b \in \mathbb{Q}$ .

## Définition ((a, b)-substitutivité)

*Soient  $a, b \in \mathbb{Q}$ . Etant donné deux automates avec coût de service  $\mathcal{A}_1$  et  $\mathcal{A}_2$  représentant respectivement deux services Web  $S_1$  et  $S_2$ , on dit que  $S_1$  peut être (a, b)-substitué par  $S_2$  si :*

- 1  $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ , et
- 2  $\forall u \in L(\mathcal{A}_1), \text{cost}_{\mathcal{A}_2}(u) \leq a \cdot \text{cost}_{\mathcal{A}_1}(u) + b$

# (a,b)-substitutivité

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre si :

- le remplaçant fournit au moins les mêmes services que le service Web à remplacer, et
- les coûts des deux services Web sont similaires considérant deux valeurs  $a, b \in \mathbb{Q}$ .

## Définition ((a, b)-substitutivité)

Soient  $a, b \in \mathbb{Q}$ . Etant donné deux automates avec coût de service  $\mathcal{A}_1$  et  $\mathcal{A}_2$  représentant respectivement deux services Web  $S_1$  et  $S_2$ , on dit que  $S_1$  peut être (a, b)-substitué par  $S_2$  si :

- 1  $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ , et
- 2  $\forall u \in L(\mathcal{A}_1), \text{cost}_{\mathcal{A}_2}(u) \leq a \cdot \text{cost}_{\mathcal{A}_1}(u) + b$

# Décidabilité de la $(a, b)$ -substitutivité

L'inclusion de langages est décidable.

L'équivalence de deux automates max-plus est décidable si ils sont finiment ambigus [Web94, HIJ02].

## Théorème

*La  $(a,b)$ -substitutivité est décidable si  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sont finiment ambigus.*

En pratique, les automates générés à partir de descriptions BPEL sont généralement non-ambigus.

# Décidabilité de la $(a, b)$ -substitutivité

L'inclusion de langages est décidable.

L'équivalence de deux automates max-plus est décidable si ils sont finiment ambigus [Web94, HIJ02].

## Théorème

*La  $(a,b)$ -substitutivité est décidable si  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sont finiment ambigus.*

En pratique, les automates générés à partir de descriptions BPEL sont généralement non-ambigus.

# Décidabilité de la $(a, b)$ -substitutivité

L'inclusion de langages est décidable.

L'équivalence de deux automates max-plus est décidable si ils sont finiment ambigus [Web94, HIJ02].

## Théorème

*La  $(a,b)$ -substitutivité est décidable si  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sont finiment ambigus.*

En pratique, les automates générés à partir de descriptions BPEL sont généralement non-ambigus.

# Décidabilité de la $(a, b)$ -substitutivité

L'inclusion de langages est décidable.

L'équivalence de deux automates max-plus est décidable si ils sont finiment ambigus [Web94, HIJ02].

## Théorème

*La  $(a,b)$ -substitutivité est décidable si  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sont finiment ambigus.*

En pratique, les automates générés à partir de descriptions BPEL sont généralement non-ambigus.

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre et inversement.

Définition à partir de la définition de la  $(a, b)$ -substitutivité.

Egalité de langages et proportionnalité des coûts.

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre et inversement.

Définition à partir de la définition de la  $(a, b)$ -substitutivité.

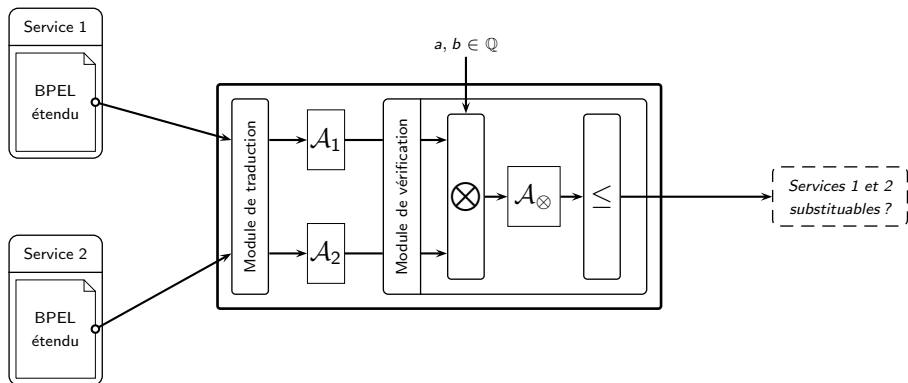
Egalité de langages et proportionnalité des coûts.

Notion permettant de dire qu'un service Web donné peut être remplacé par un autre et inversement.

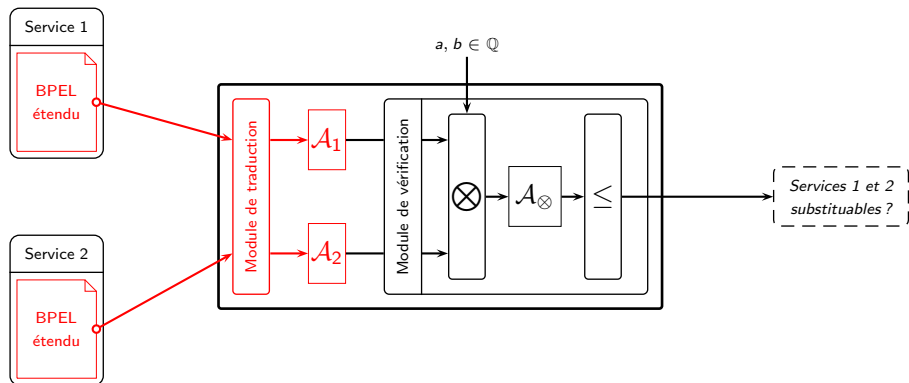
Définition à partir de la définition de la  $(a, b)$ -substitutivité.

Egalité de langages et proportionnalité des coûts.

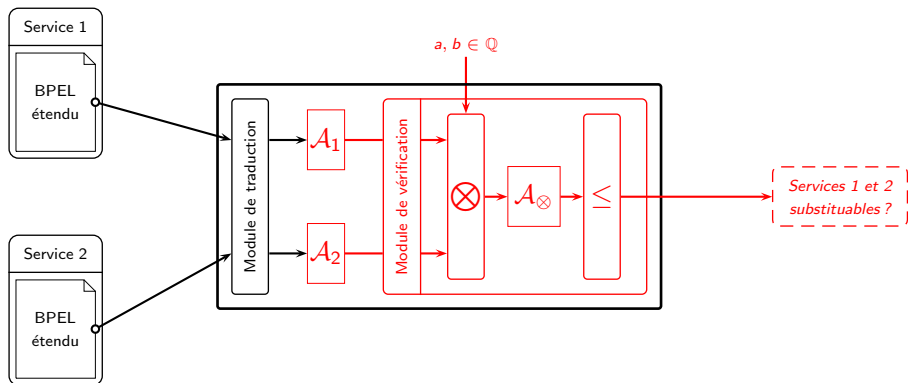
# Implémentation de l'approche



# Implémentation de l'approche



# Implémentation de l'approche



## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la (a,b)-substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.

## Perspectives théoriques :

- Définition d'une notion de compatibilité requêtes/services.
- Utilisation d'autres relations d'équivalence à la place de l'équivalence de trace.
- Extension des automates avec coût de service avec des gardes sur les transitions.

## Perspectives pratiques :

- Intégration des différentes notions au module de vérification.
- Amélioration de l'implémentation de l'algorithme de vérification de la  $(a,b)$ -substitutivité.
- Développement d'un module de traduction pour une version étendue de OASIS WS-BPEL 2.0.



Ivona Brandic, Sabri Pilana, and Siegfried Benkner.  
High-level Composition of QoS-aware Grid Workflows: An Approach that Considers Location Affinity.  
*In Proc. of the HPDC-15 Workshop on Workflows in Support of Large-Scale Science (WORKS'06), Paris, France, 2006.*







Antonella Chirichiello and Gwen Salaun.  
Formal Development of Web Services.  
*In Proc. of the 4th International Workshop on AI for Service Composition (AISC'06), Trento, Italy, 2006.*



Howard Foster.  
*A Rigorous Approach to Engineering Web Service Compositions.*  
PhD thesis, Imperial College London, 2006.

-  Rachid Hamadi and Boualem Benatallah.  
A Petri Net-based Model for Web Service Composition.  
*In Proc. of the 14th Australasian Database Conference (ADC'03), Adelaide, Australia, 2003.*
-  Kosaburo Hashiguchi, Kenichi Ishiguro, and Shuji Jimbo.  
Decidability of the Equivalence Problem for Finitely Ambiguous Finance Automata.  
*IJAC, 12(3), 2002.*
-  Alexander Keller and Heiko Ludwig.  
The WSLA Framework: Specifying and Monitoring of Service Level Agreements for Web Services.  
Technical Report IBM RC22456, 2002.

## Références III

-  Akhil Sahai, Anna Durante, and Vijay Machiraju.  
Towards Automated SLA Management for Web Services.  
Technical Report Research report HPL-2001-310 (R.1), 2002.
-  Ali Shaikh Ali, Omer F. Rana, Rashid J. Al-Ali, and David W. Walker.  
UDDIe: An Extended Registry for Web Service.  
In *Proc. of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03), Orlando, FL, USA, 2003.*
-  Min Tian.  
*QoS integration in Web services with the WS-QoS framework.*  
PhD thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, 2005.
-  Andreas Weber.  
Finite-valued Distance Automata.  
*TCS*, 134, 1994.