Modélisation et vérification d'un ou plusieurs services web.

RGE, à Besançon

19 octobre 2006

Sylvain Rampacek
sylvain.rampacek@univ-reims.fr

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

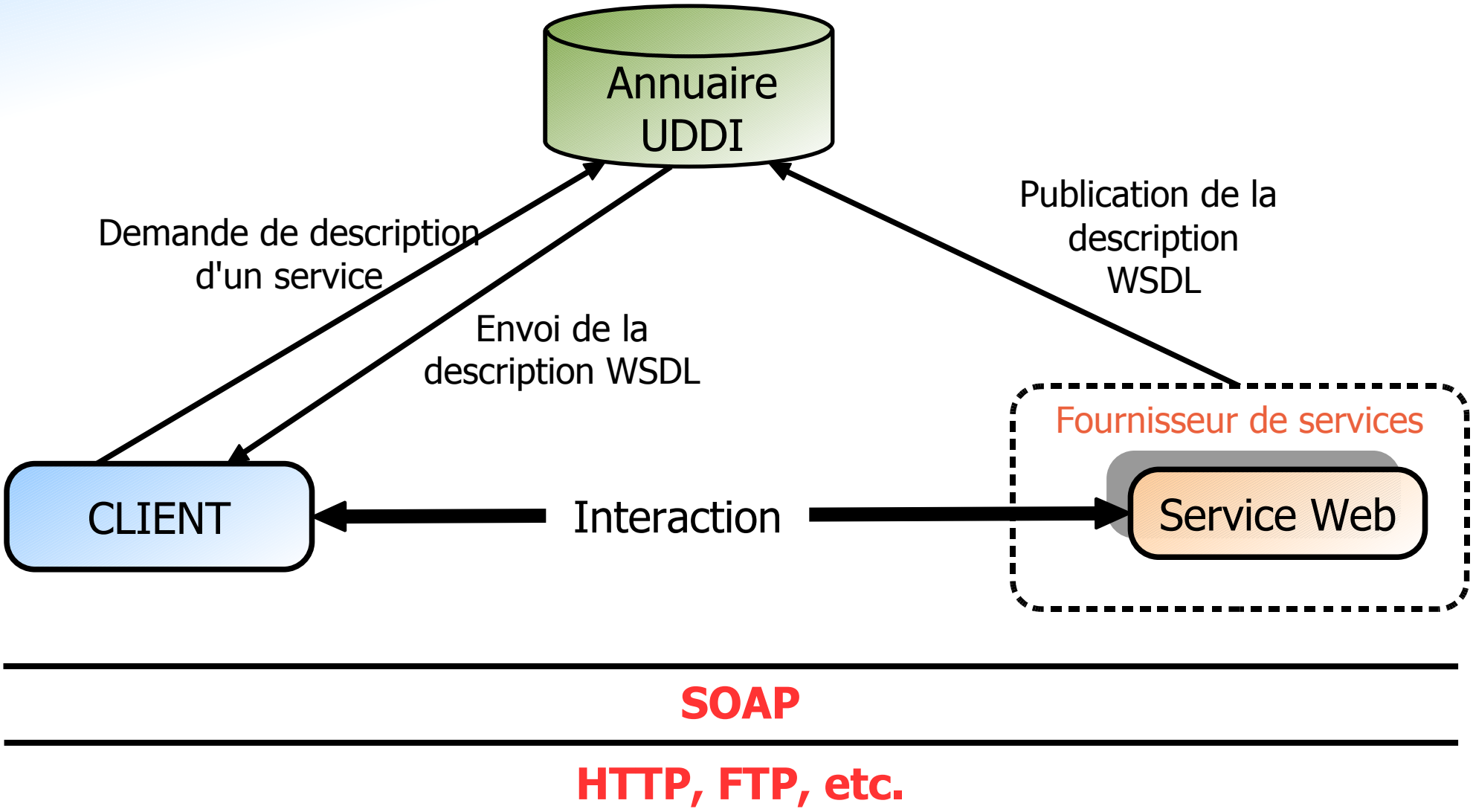
Laboratoire CReSTIC
Équipe SysCom

Plan

- **1. Introduction**
- **2. Formalisation et Modélisation**
  - Serveur
  - Client
- **3. Vérification d'une chorégraphie**
  - Partenaires
  - Agrégations
- **4. Conclusion**

1. Introduction

Architecture des Services Web

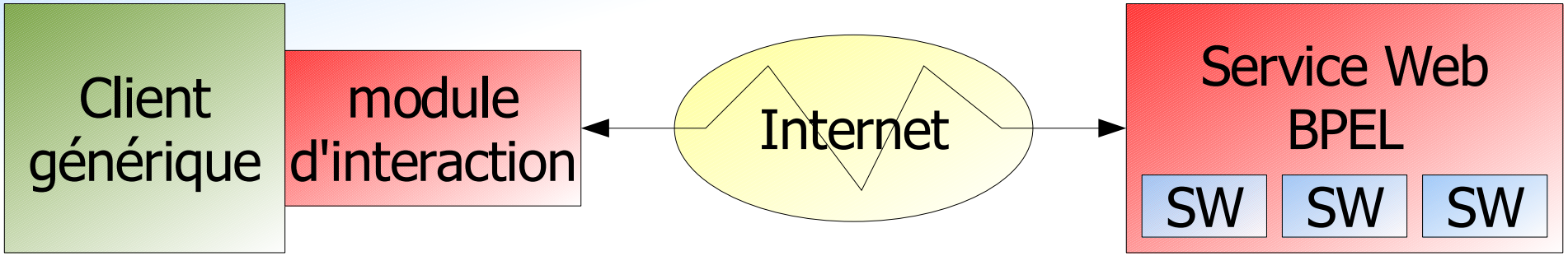


Services web : protocoles et langages

technologies XML

- **Protocole de communication : SOAP**
  - représente les données
  - description des noeuds intermédiaires (routage)
- **Langage de description : WSDL**
  - niveau interface du service
  - descriptions des types, paramètres et méthodes accessibles
- **Langage de description comportementale : BPEL**
  - décrit le processus métier
  - articulation entre les différentes méthodes
  - nombreux exemples : WSFL, XLANG, BPEL

Problématique



- **Problèmes soulevés**
  - **Formalisation** et **modélisation** des services web
    et génération d'un modèle client adapté
  - *L'**interaction** (au sens algorithmique) est-elle possible avec ce service ?*
    si ce n'est pas le cas : détection de l'**ambiguïté**
  - Service web composé de **plusieurs sous-services ?**
    => vérifier l'interaction entre ces sous-services

2. Formalisation et Modélisation

2. Formalisation et Modélisation

Introduction

- **Objectifs :**
  - obtenir un modèle du service...
  - ... permettant la génération du modèle client
  - indiquer les cas d'ambiguïté
- **Formalisation** du langage de description comportementale
  - utilisation de modèle reflétant l'interaction
- **Approches *temps discret* et *temps dense***
  - Algèbre de processus temporisés (Sifakis)
  - TIOTS (système de transitions)
  - Automates Temporisés (Alur et Dill)

Sémantique temporisée d'un processus BPEL

actions et règles (temps dense)

Actions d'un service web

- envoi/réception d'un message — $!o[m]$ / $?o[m]$
- action interne (immédiate, côté service) — τ
- exception — e
- action de terminaison — $\surd$
- expiration d'un *time-out* — to

Règles de la sémantique

processus *empty*

$$empty \xrightarrow{\surd} 0$$

processus *operation*

$$*o[m] \xrightarrow{*m} empty$$

processus *while*

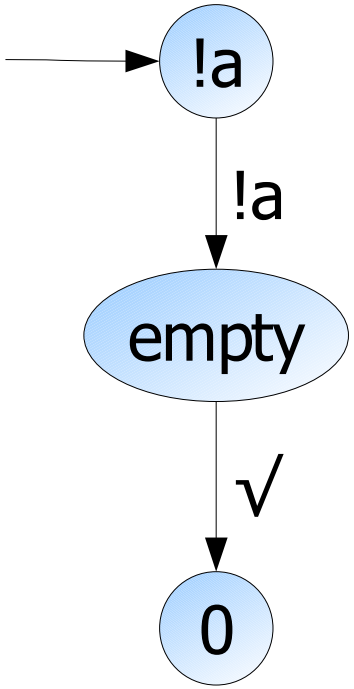
$$while[P] \xrightarrow{\tau} empty$$

$$while[P] \xrightarrow{\tau} P\,;while[P]$$

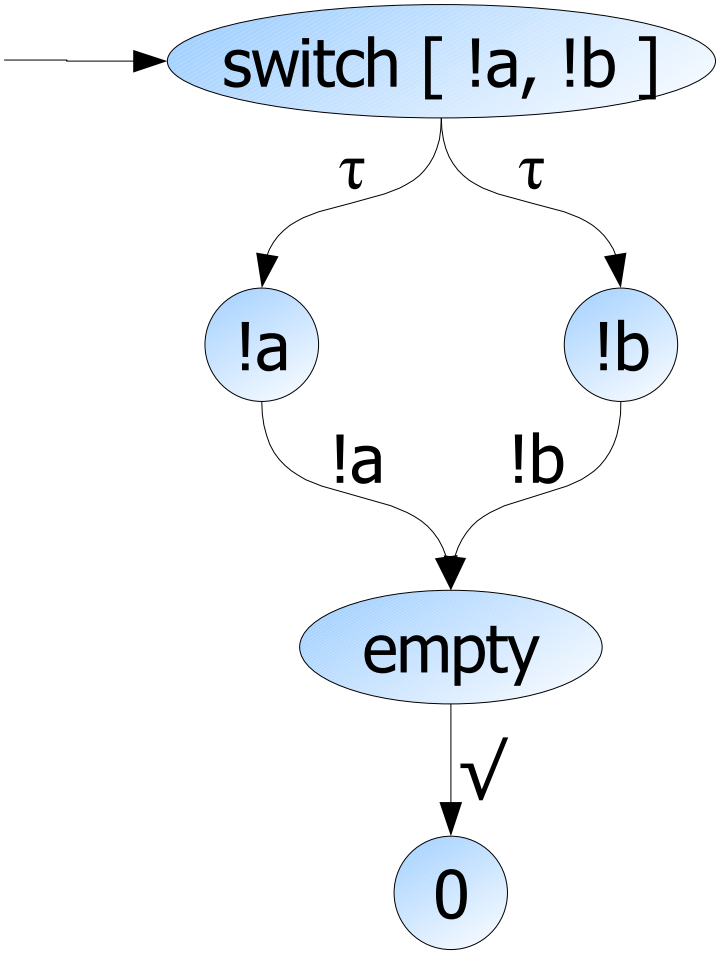
Synthèse de l'AT du service

(exemples sans horloge)

!a



switch[!a, !b]



la synthèse de l'AT : application récursive des règles

Relation d'interaction

Quel client pour un service ?

- **Le client doit :**
  - être un programme **déterministe**
  - communiquer en envoyant et recevant des messages
  - gérer le passage du temps
- donc : représentation par un **automate temporisé**

- **A tout instant de l'interaction**, vérifier les règles concernant **:**
  - les messages reçus
  - les messages envoyés
  - les contraintes d'horloges
  - la détection de la **terminaison** de l'interaction.

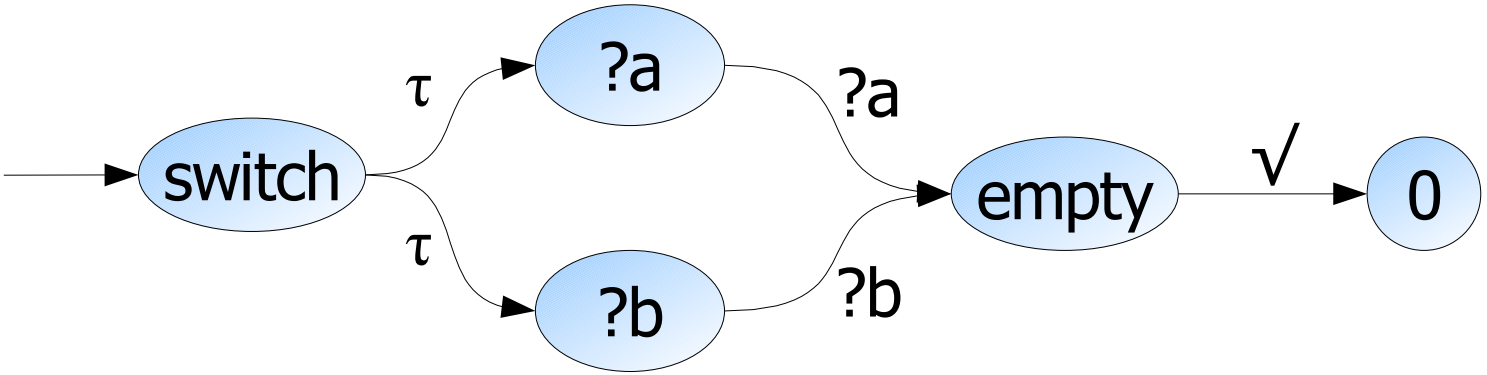
Il y a **ambiguïté** si un tel client n'existe pas

Ambiguïté d'un service

exemple

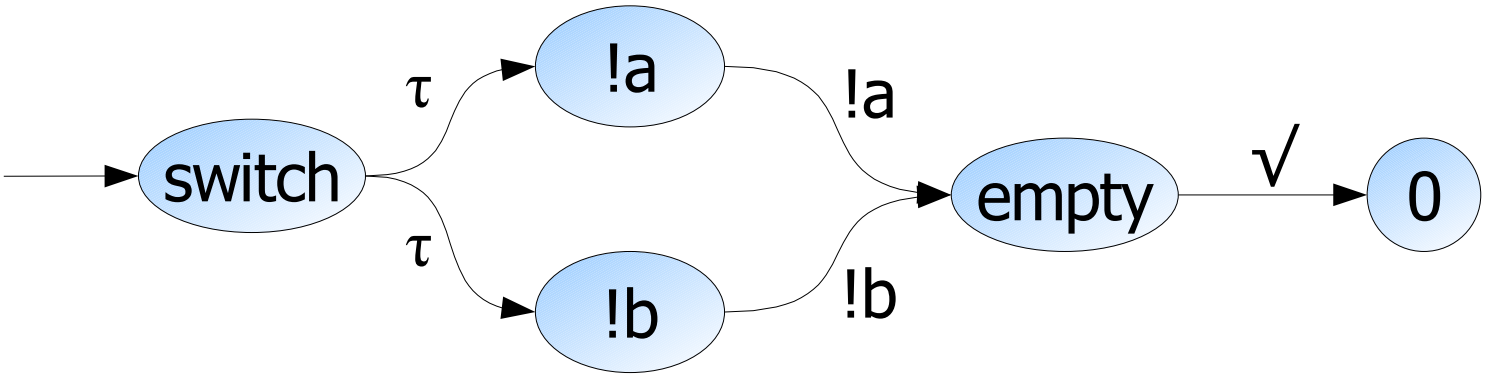
switch [?a , ?b]

- **ambigu :** car le client ne sait pas quel message envoyer



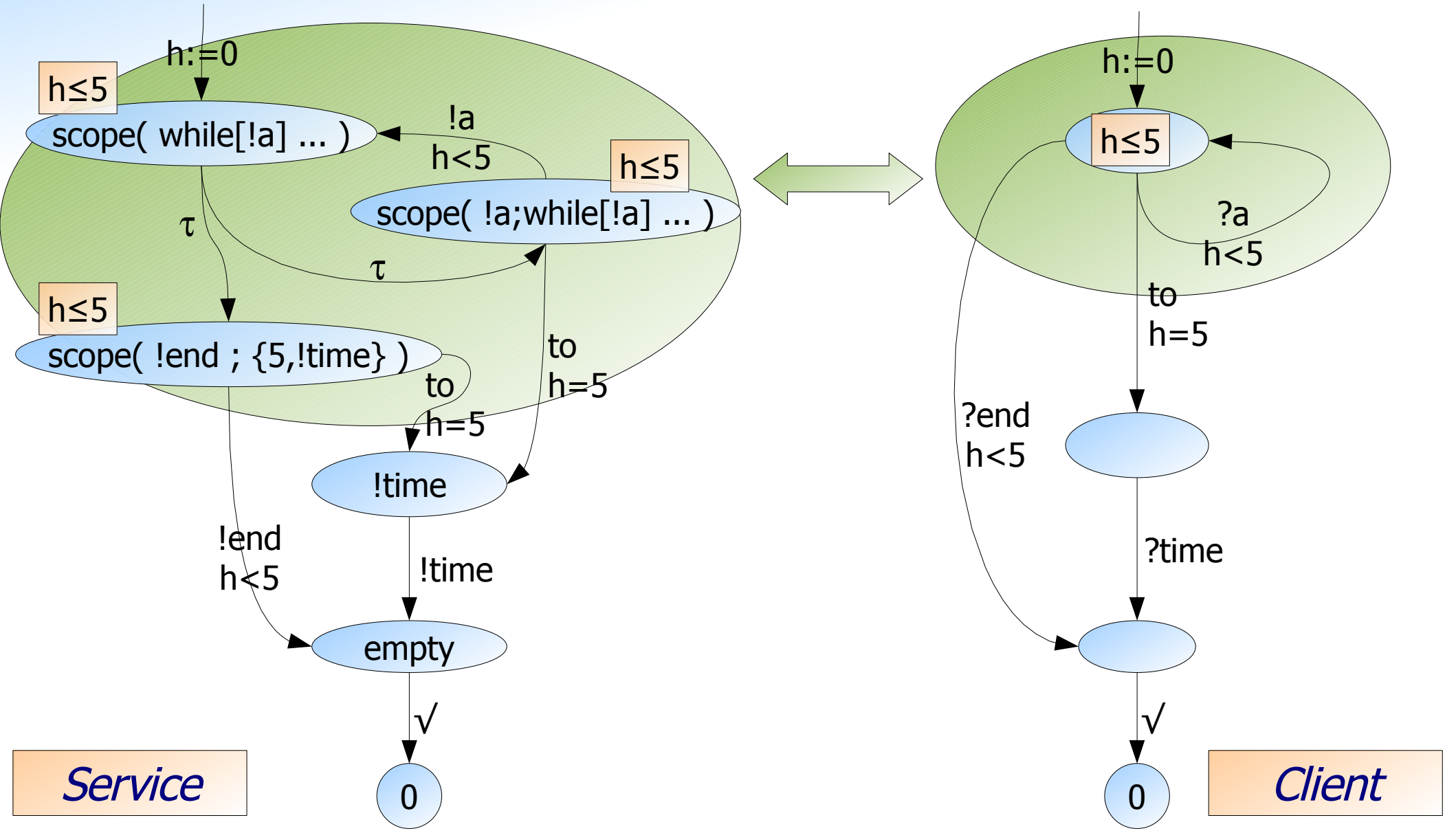
switch [!a , !b]

- **non ambigu :** car le client attendra l'un ou l'autre des messages



Exemples d'automates du service et du client

scope(while[!a];!end , {5, !time})



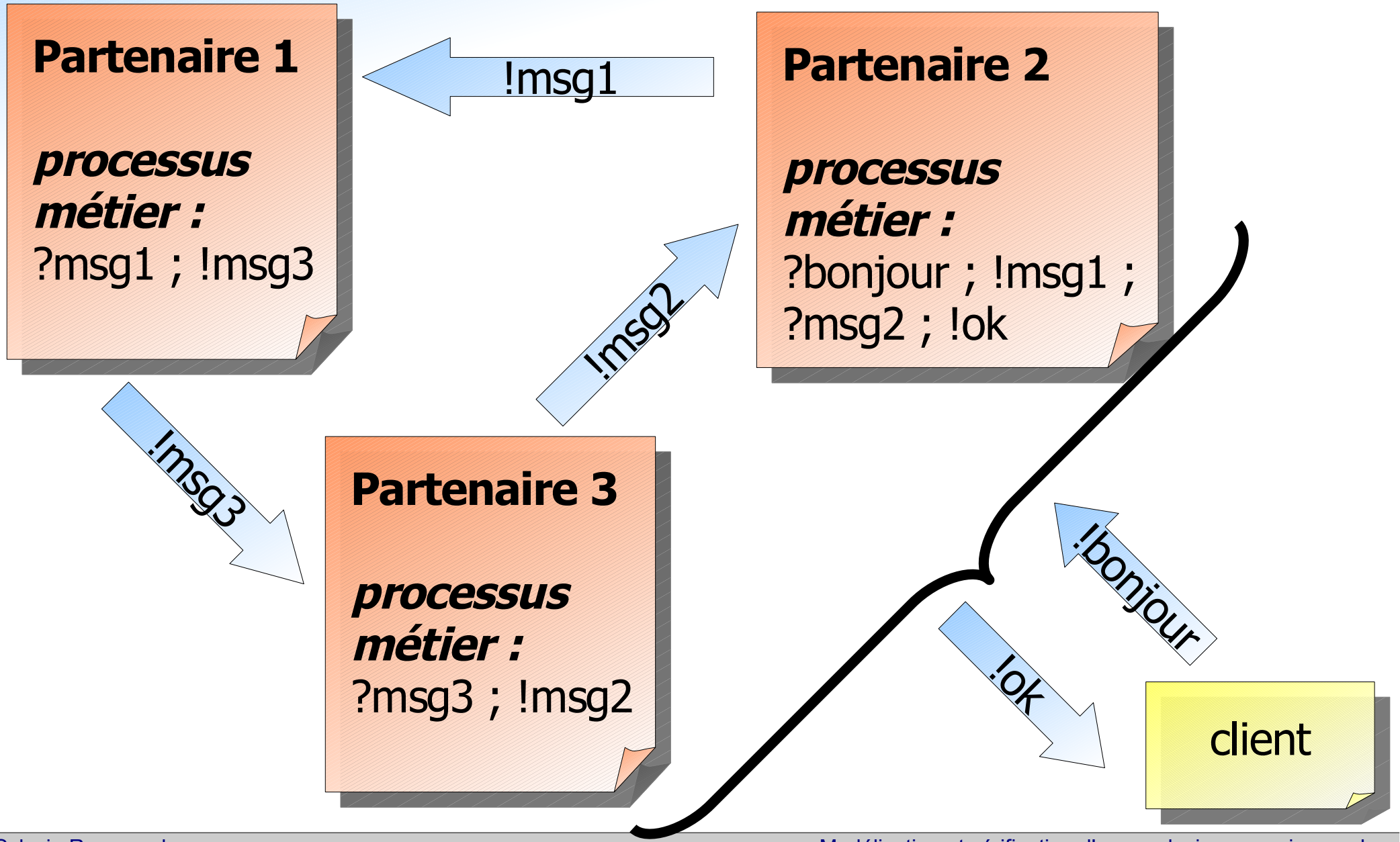
3. Vérification d'une chorégraphie

3. Vérification d'une chorégraphie

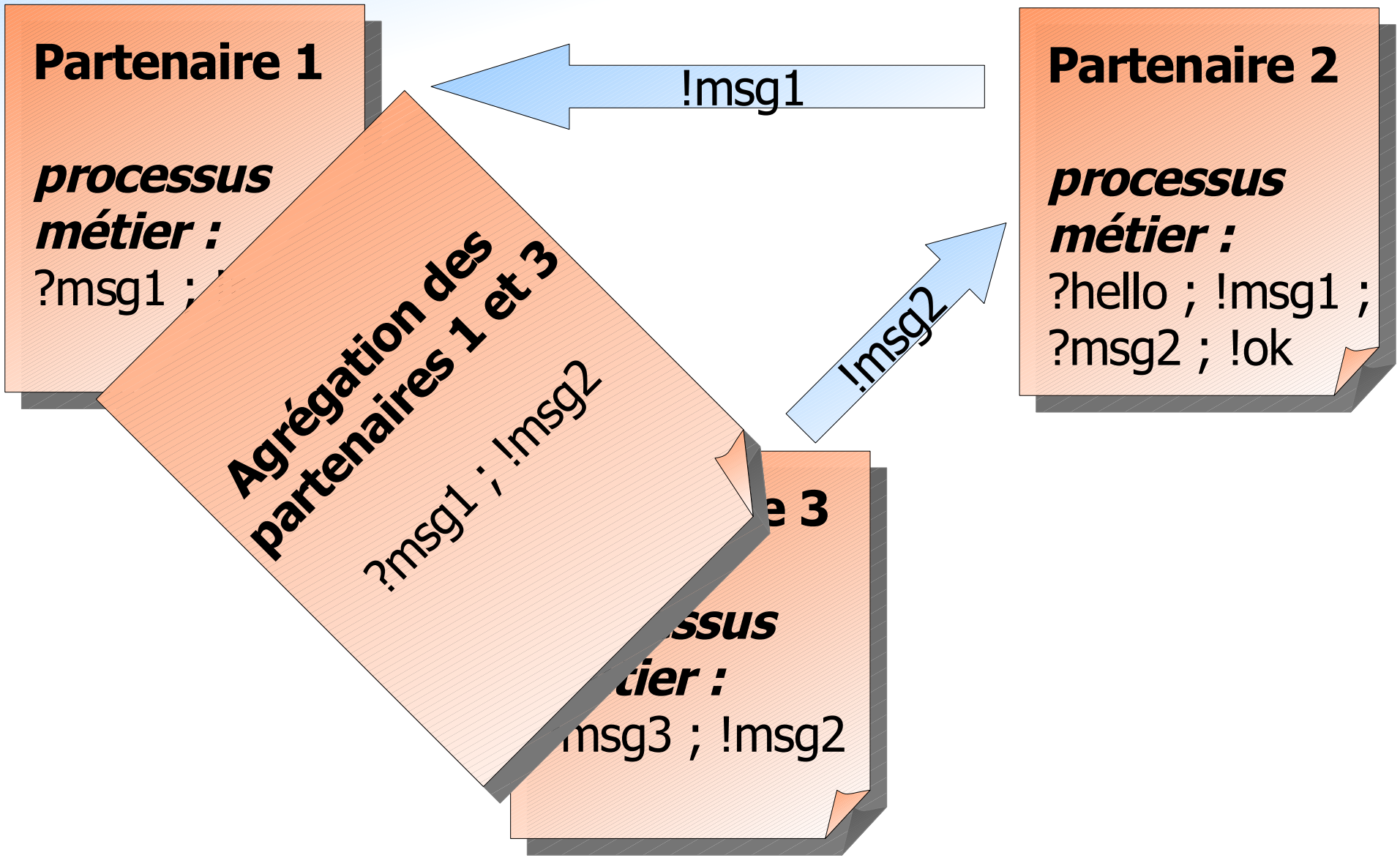
Introduction

- **Définitions**
  - **orchestration :** un coordinateur central
  - **chorégraphie :** absence de coordinateur central
- **Extension de la méthode précédente ...**
  - ... à un ensemble de services évoluant dans une chorégraphie
- Importance de la **source** et du **destinataire** du message
- Les modèles sont obtenus par la description des processus métier des **(sous-)services web**
  - sans génération de modèle représentant un client adapté

Chorégraphie et Partenaires



Agrégation de partenaires



Vérification d'une chorégraphie

- Consiste à vérifier l'**absence d'ambiguïté** dans l'**interaction interne** à la chorégraphie

- Chaque partenaire, en parallèle
  - **agrège** le comportement des autres partenaires
  - puis **vérifie** l'interaction avec cette agrégation à l'aide de l'adaptation des algorithmes précédents

4. Conclusion

Conclusion

- **Formalisation et Modélisation**
  - formalisation de langages de description comportementale
  - modélisation du service et génération d'un modèle client adapté
  - détection de l'ambiguïté d'un service par le biais de la relation d'interaction
  - aspect générique de la modélisation
- **Résultats Technologiques**
  - client générique
  - outils s'intégrant à une plateforme « services web »

Bibliographie

- [HMR06] S. Haddad, P. Moreaux, and S. Rampacek. Client synthesis for web services by way of a timed semantics (ICEIS06).
- [BMR06] C. Boutrous-Saab, T. Melliti et S. Rampacek. *Verifying correctness of web services choreography* (ECOWS06) – *à paraître* –

Perspectives

- **Étendre l'aspect vérification d'une chorégraphie**
  - sémantique temps dense
  - vers la composition automatique ou le remplacement automatique de services
- Adapter notre approche à d'**autres domaines** que les services web