Prise en compte d'assertions pour la correction d'assemblages de composants Kmelia

Pascal ANDRE Gilles ARDOUREL Christian ATTIOGBE Arnaud LANOIX
Mohamed MESSABIHI

COLOSS / LINA – UMR CNRS 6241
{Prenom.Nom}@univ-nantes.fr

Journée thématique COSMAL
Journées Nationales du GDR GPL

Plan

Kmelia : un modèle à composants multi-services

[Attiogbé et al., 2006]

Composant abstrait, non exécutable

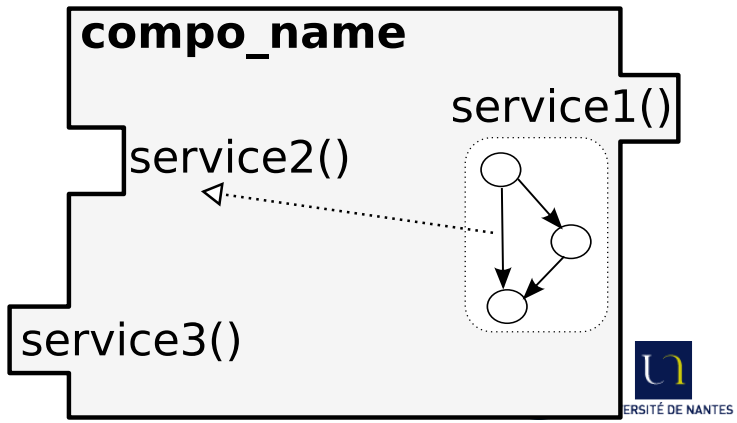
- Espace d'états = variables + constantes
- *Invariant*
- Interface = Services requis + offerts

Service "fonctionnalité" du composant

- Variables locales
- Interface = sous-services (internes ou requis/offerts)
- *Assertions* = *pré-/postconditions*
- Comportement dynamique = eLTS

Assemblage liens d'assemblage entre services offerts/requis

```
Component compo_name
    Interface    <Interface descr.>
    Types        < Type Defs >
    Variables    <Var list>
    Invariant    <Predicate>
    Initialisation
        ...      // var. assignments
    Services
        ...
end
```



Kmelia : un modèle à composants multi-services

[Attiogbé et al., 2006]

Composant abstrait, non exécutable

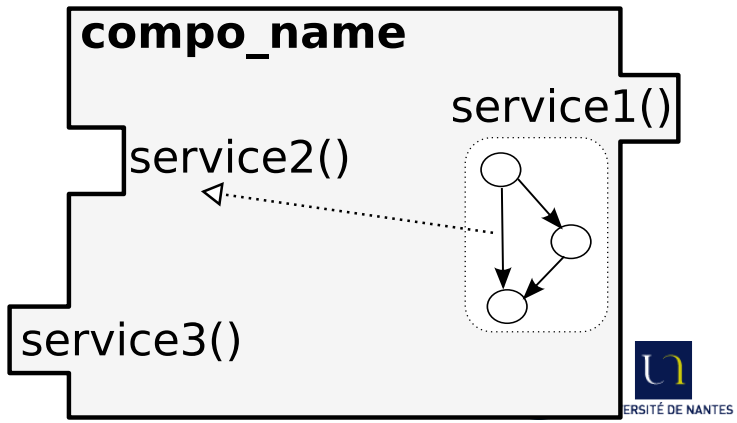
- Espace d'états = variables + constantes
- *Invariant*
- Interface = Services requis + offerts

Service "fonctionnalité" du composant

- Variables locales
- Interface = sous-services (internes ou requis/offerts)
- *Assertions* = *pré-/postconditions*
- Comportement dynamique = eLTS

Assemblage liens d'assemblage entre services offerts/requis

```
Provided service_1 ()
    Interface   <Interface descr>
    Pre         <Predicate>
    Post        <Predicate>
    Behaviour
        init    q_0
        final   q_f
        { ...,
        q_i - - label - - > q_j,
        ... }
end
Required service_2 () ...
```



Kmelia : un modèle à composants multi-services

[Attiogbé et al., 2006]

Composant abstrait, non exécutable

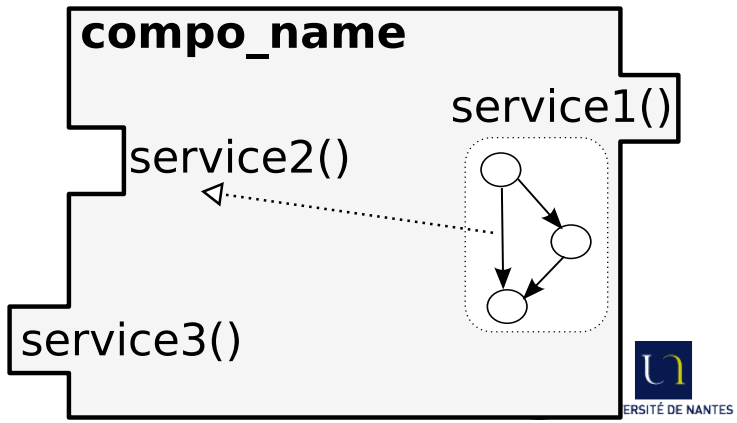
- Espace d'états = variables + constantes
- *Invariant*
- Interface = Services requis + offerts

Service "fonctionnalité" du composant

- Variables locales
- Interface = sous-services (internes ou requis/offerts)
- *Assertions = pré-/postconditions*
- Comportement dynamique = eLTS

Assemblage liens d'assemblage entre services offerts/requis

```
Provided service_1 ()
    Interface   <Interface descr>
    Pre         <Predicate>
    Post        <Predicate>
    Behaviour
        init    q_0
        final   q_f
        { ...,
        q_i - - label - - > q_j,
        ... }
end
Required service_2 () ...
```



Vérification d'un modèle Kmelia

Service

- Typage
- *Cohérence du eLTS vis-à-vis des pré-/postconditions*
- *Correction des appels de services*

Composant

- Typage
- *Respect de l'invariant par les pré-/post des services*
- *Correction des enchaînements autorisés de services (service "Protocole")*

Assemblage

- Signature (simple) des services
- Signature "profonde" = signature des sous-services
- *Contrat d'assemblage = pré-/post requises vs. offertes*
- Vérification dynamique = interaction des eLTS

Vérification d'un modèle Kmelia

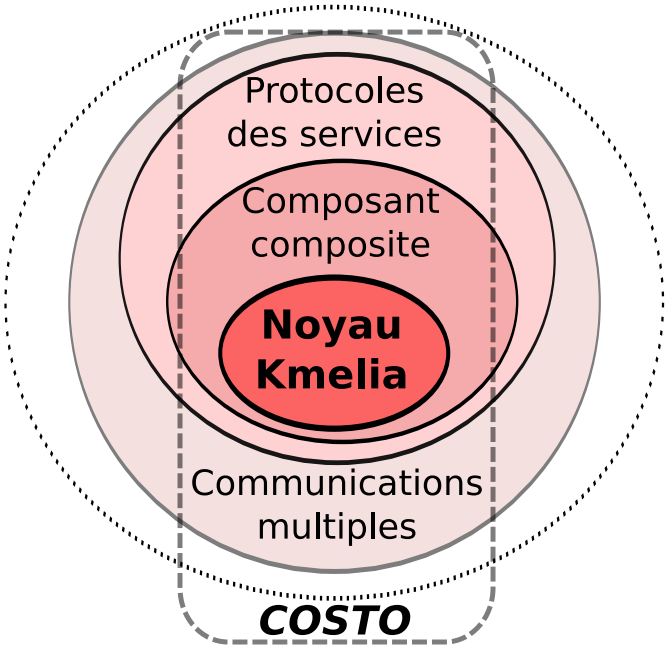
Service
- Typage
- *Cohérence du eLTS vis-à-vis des pré-/postconditions*
- *Correction des appels de services*

Composant
- Typage
- *Respect de l'invariant par les pré-/post des services*
- *Correction des enchaînements autorisés de services (service "Protocole")*

Assemblage
- Signature (simple) des services
- Signature "profonde" = signature des sous-services
- *Contrat d'assemblage = pré-/post requises vs. offertes*
- Vérification dynamique = interaction des eLTS

Construction incrémentale de Kmelia

- Noyau Kmelia [Attiogbé et al., 2006]
  - Primitives stables : composant, services, eLTS, ...
  - Méthodologie pour la vérification
- Extensions
  - Composant "composite" / promotion de services [André et al., 2006]
  - Protocoles d'usage des services [André et al., 2007]
  - Services partagés et communications multiples [André et al., 2008]
  - ...



- Parallèlement,
  développement d'une plateforme d'expérimentation "ouverte" : COSTO
  - Type-checker pour les spécifications Kmelia
  - Passerelles vers MEC et LOTOS/CADP : vérification des interactions entre eLTS
  - *Passerelle vers B (en cours)*

Plan

Enrichissement du langage de données de Kmelia

Motivation décrire des applications "réelles" en Kmelia

Objectif pouvoir exprimer/utiliser l'invariant du composant et les pré/post des composants + des propriétés

Contrainte compromis entre l'expressivité (souhaitée) et les contraintes de l'existant : eLTS, communications

Types Types de base usels : `Integer` | `Boolean` | `Char` | `String`

`struct(`Type1`,` Type2`,...)` | `array[`n`]` `of` Type

`enum{` e1`,...,` en`}` | `setOf` Type

Expressions constantes, variables

opérateurs classiques : `+` | `*` | `mod` | `<` | `>=` | `!=` | `...`

Assertions `Not(`pred`)` | `==>` | `\exists` | `\forall`

Plan

Assertions pour la vérification

Motivation Assurer qu'une spécification Kmelia est correcte

Service
- *Cohérence du eLTS vis-à-vis des pré-/post du service*
- Correction des appels de services ▸ Détails

Composant
- Respect de l'invariant par les pré-/post des services
- Correction des enchaînements autorisés de services (service "Protocole") [André et al., 2007]

Assemblage
- Contrat d'assemblage = pré-/post requises vs. offertes ▸ Détails

$\Rightarrow$ Définition des Obligations de Preuve correspondantes à chaque niveau

Expérimentation Intégration dans COSTO
- Réutilisation d'outils de preuve existants : B, Z, Coq...
- Lien avec des travaux précédents sur l'utilisation de B pour l'adaptation de composants [Lanoix et al., 2008]

Plan

Conclusion et perspectives

- Enrichissement de Kmelia pour prendre en compte des assertions pour la correction d'assemblage de composants [André et al., 2009]
  - Définition "précise" d'un langage d'assertions (type de base, logique, expressions, . . .)
  - Définition des différentes procédures de vérification associées à chaque niveau : service, composant, assemblage

- Perspectives directes :
  - Définir "plus précisément" certaines procédures de vérification
  - Mise en oeuvre des différentes procédures dans COSTO (actuellement, expérimentations avec B)

- Plus généralement
  - Composants paramétrés / assemblage paramétré
  - Schémas d'assertions pour aider l'utilisateur
  - Expression/vérification de propriétés non-fonctionnelles en Kmelia
  - Visibilité des variables (services/composants) dans les composites
  - Composants vs. Aspects
  - . . .

Références I

André, P., Ardourel, G., and Attiogbé, C. (2006).
Spécification d'architectures logicielles en Kmelia : hiérarchie de connexion et composition.
In *1ère Conférence Francophone sur les Architectures Logicielles*, pages 101–118. Hermès, Lavoisier.

André, P., Ardourel, G., and Attiogbé, C. (2007).
Defining Component Protocols with Service Composition : Illustration with the Kmelia Model.
In *6th International Symposium on Software Composition, SC'07*, volume 4829 of *LNCS*. Springer.

André, P., Ardourel, G., and Attiogbé, C. (2008).
Composing Components with Shared Services in the Kmelia Model.
In *7th International Symposium on Software Composition, SC'08*, volume 4954 of *LNCS*. Springer.

André, P., Attiogbé, C., and Messabihi, M. (2009).
Correction d'assemblages de composants impliquant des interfaces paramétrées.
In *3ième Conférence Francophone sur les Architectures Logicielles*. Hermès, Lavoisier.

Attiogbé, C., André, P., and Ardourel, G. (2006).
Checking Component Composability.
In *5th International Symposium on Software Composition, SC'06*, volume 4089 of *LNCS*. Springer.

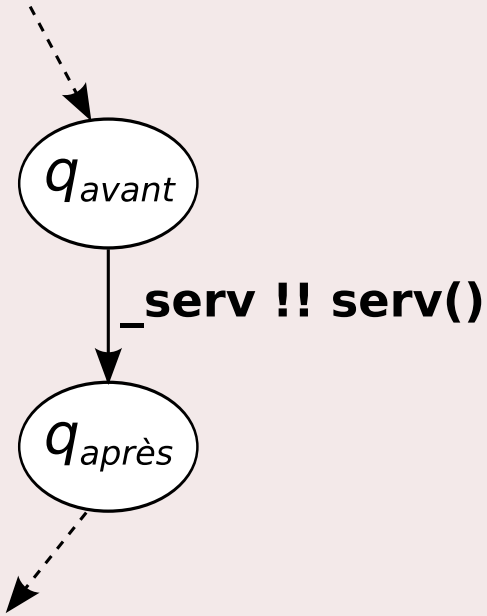
Lanoix, A., Colin, S., and Souquières, J. (2008).
Développement formel par composants : assemblage et vérification à l'aide de B.
Technique et Science Informatiques (TSI), 26(8) :1007–1032.
Numéro spécial AFADL07.

Correction des appels de services

Appel de service

L'appel à $< pre_{serv} > serv() < post_{serv} >$ depuis un état q_{avant} (et conduisant à un état q_{apres}) est correct ssi

1. $pred(q_{avant}) \Rightarrow pre_{serv}$
2. $post_{serv} \Rightarrow pred(q_{apres})$



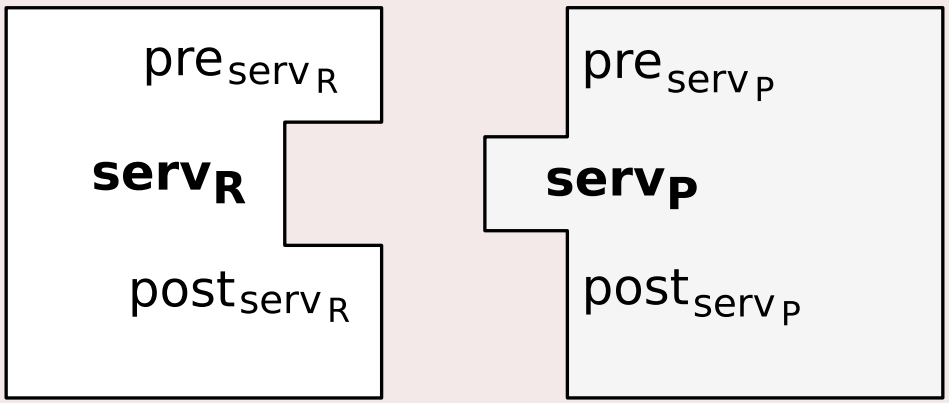
Retour

Correction des contrats d'assemblage

Contrat d'assemblage

Un service requis $< pre_{serv_R} > serv_R() < post_{serv_R} >$ peut être connecté avec un service fourni $< pre_{serv_P} > serv_P() < post_{serv_P} >$ ssi

1. $pre_{serv_R} \Rightarrow pre_{serv_P}$
2. $post_{serv_P} \Rightarrow post_{serv_R}$



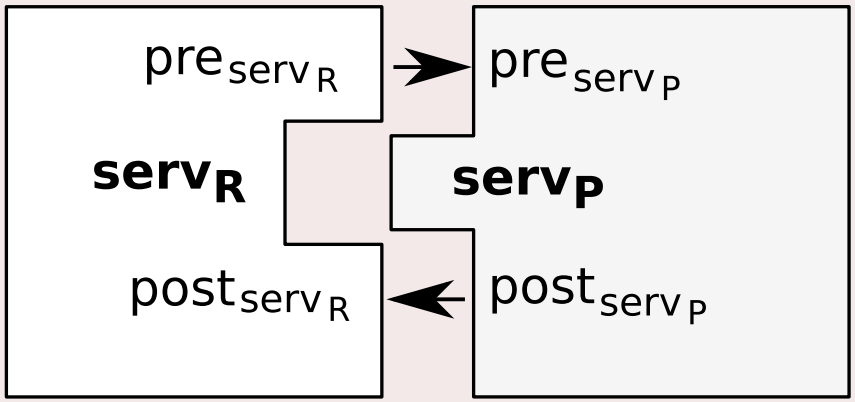
- La vérification des liens de promotion est similaire.

Retour

Contrat d'assemblage

Un service requis $< pre_{serv_R} > serv_R() < post_{serv_R} >$ peut être connecté avec un service fourni $< pre_{serv_P} > serv_P() < post_{serv_P} >$ ssi

1. $pre_{serv_R} \Rightarrow pre_{serv_P}$
2. $post_{serv_P} \Rightarrow post_{serv_R}$



- La vérification des liens de promotion est similaire.

Retour