Live Goals for Adaptive Service Compositions

Luciano Baresi - Liliana Pasquale

Politecnico di Milano
The SOA paradigm is intrinsically self-adaptive
- BPEL: technical solutions for monitoring/adaptation

When
adaptation must be activated at runtime

What
Adaptation capabilities are needed

How
enrich BPEL processes with self/adaptation capabilities

Motivations
Our solution

Funtional/non-functional requirements

KAOS

SEAMS 2010
L. Baresi, L. Pasquale – Live Goals for Adaptive Service Compositions
Our approach by example

Example

- Model an application for organizing dinners

Requirements

- Book a restaurant that matches users’ preferences (cost)

Adaptation capabilities

- Try to meet user preferences when possible.
- Otherwise choose the cheapest solution (takeaways)
KAOS goals

Entities:
- Rest_List
- Sel_Restaurant
- Restaurant
- Preference

Agents:
- a1
- a2

Operations:
- Op1 [Get preferences]
- Op2 [Show possibilities]
- Op3 [Select restaurant]
- Op4 [Send booking request]
- Op5 [Get confirmation]
- Op6 [Update arrival time]
- Op7 [Notify cancellations]

Domain Precondition:
- l.state = empty

Domain Postcondition:
- l.state = filled

Trigger Precondition:
- (p.state = filled)

Requirement Postcondition:
- show possibilities

Entities Precondition:
- r : Restaurant
  - r.id ≠ null
  - r.name ≠ ""
  - r.address ≠ ""

Entities Postcondition:
- r : Restaurant
  - r.id = s.restaurant

States:
- l : Rest_List
- p : Preference
- s : Sel_Restaurant

Conditions:
- l.state = filled
- p.state = filled
- s.state = selected

Environments:
- t : Time
- y : Year

Variables:
- r : Restaurant
  - r.id
  - r.name
  - r.address

Logic:
- l.state = filled
- p.state = filled
- s.state = selected
- r.id = s.restaurant
- r.name ≠ ""
- r.address ≠ ""
- r.id ≠ null
- r.name ≠ ""
- r.address ≠ ""
Adaptive goals

Change Agent (Op2, …)

AG1.2.1
[Price complies with preferences]

G1.2
[Choose restaurant]

Op2′
Op3′

Agent: a2
Entities: Preference, Place_List, Restaurant'

Preferences are not met

Required price too low

Conditions
p : Preference(p.price ≤ 20)

Objectives:
Enforce weaker version of goal G1.2.1

Objectives:
Enforce goal G1.2

l : Place List,
p : Preference (@(l.state = filled) ⇒ ∀ r : Restaurant (r ∈ l ∧ r.type = take away))

Remove Entity (Rest_List)

Remove Operation (Op2, Op3)

Op2
Op3

S2

S1

Remove Goal (G1.2)

Add Entity (Place_List)

Modify Entity (Restaurant)

Add Operation (Op2′, Op3′)
1. Lifecycle of each entity:
   • state transitions
   • operations causing state transitions
   • precedence rules among operations

Rest_List

Op2
empty
filled

Op1 < Op2
DomPost(Op1) ⇒ DomPre(Op2)
ReqPost(Op1) ⇒ TrigPre(Op2)

Op2 < Op3
DomPost(Op2) ⇒ DomPre(Op3)
ReqPost(Op2) ⇒ TrigPre(Op3)
2. Generate a possible operation sequence

3. Associate entities with process variables:
   Preference $\rightarrow p$
   Rest_List $\rightarrow l$
   Sel_Restaurant $\rightarrow s$

1. Map agents to partner services:
   $a1 \rightarrow p1$, $a2 \rightarrow p2$
5. Operations and events are translated into concrete process activities.

**Regenerated Postconditions**
- refers to an event
- changes an entity

A set of assign activities is generated, optionally preceded by a Invoke(-Receive) activity.

**Required Postconditions**
- contains an event

It is translated into an Invoke, Invoke/Receive, or Reply activity.

---

**Name:** Op3

**In/Out:** p : Preference, l : Rest_List

**DomPre:** l.state = empty

**DomPost:** l.state = filled

**TrigPre:** @(p.state = filled)

**ReqPost:** show possibilities

\[
\forall r \in l (r.id \neq \text{null} \land r.name \neq \text{""} \land r.address \neq \text{""})
\]
Supervision Model

- **Supervision directives**
  - **Monitoring directive:**
    - decide when adaptation must be carried out
    - translation of goals in the language of a monitor
      - Dynamo: FOL properties
      - Albert: LTL properties
  - **Adaptation directives:**
    - Scope
    - Conflicts with other strategies
    - Actions:
      - change the process instance/process definition
      - change the goal model
Conflicting strategies

Conflicts Types
1. Applied on the same goal with overlapping conditions
2. Applied on the benefit of conflicting goals
   • the conflict is specified in the goal model
3. May generate incoherent processes

Conflicts Resolution
1. The strongest strategy is triggered
   • strategies are defined in a hierarchical way
2. The strategy associated with the most critical goal is triggered
Example (1/2)

S1

- **Conditions**
  
  \[ p : \text{Preference}(p.\text{price} \leq 20) \]

- **Objectives:**
  
  Enforce weaker version of goal G1.2

  \[ l : \text{Place List}, \]

  \[ p : \text{Preference} (\@((l.\text{state} = \text{filled}) \Rightarrow \forall r : \text{Restaurant} (r \in l \land r.\text{type} = \text{take away})) \]

S2

- **Objectives:**
  
  Enforce goal G1.2.1

<table>
<thead>
<tr>
<th>AG1.2.1</th>
<th>Strategy</th>
<th>Scope</th>
<th>Conflicts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1</td>
<td>instance</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>instance</td>
<td>S1</td>
</tr>
</tbody>
</table>
Example (2/2)

Op1

invoke find_restaurant
(cost, time)

receive show_possibilities
(rest_list)

assign rest_list
(rest_list → l)

invoke
show_possibilities

pick get_selection
(restaurant)

assign restaurant
(restaurant → s.restaurant)

Op4

Op1

invoke find_restaurant
(time)

receive show_possibilities
(place_list)

assign rest_list
(place_list → pl)

invoke
show_possibilities

pick get_selection
(restaurant')

assign restaurant
(restaurant' → s.restaurant)

Op4

l: Rest_List
r: Restaurant

l: Place_List
r: TakeAway
Conclusions and Future Work

Conclusions

- An approach to deal with self-adaptive BPEL processes from requirements elicitation
  - Adaptive goals address adaptation and changes at goal level

(Parallel/)Future Work

- Fuzzy goals to have a notion of satisfaction level for requirements, and trigger adaptation accordingly
- Design tools to model realistic service compositions
- Improve the conflict resolution methodology
Questions