Behaviour-Aware Compositions of Things using DPWS

Javier Cubo¹, Antonio Brogi², Ernesto Pimentel¹, Félix Prado¹, Jose Antonio Martin¹

¹Dept. of Computer Science, University of Málaga, Spain  
²Dept. Of Computer Science, University of Pisa, Italy
Motivation: Adaptation in SOA

Internet

Matching:
- Signature (WSDL)
- Behaviour (BPEL)
- QoS (WS-Security,...)
Motivation: Adaptation in SOA

Alternative

Internet

Legacy

Incompatibilities:
- Deadlocks
- Livelocks
- SLA violations

Matching:
- Signature (WSDL)
- Behaviour (BPEL)
- QoS (WS-Security,...)
Motivation: Adaptation in SOA

Alternative

Internet

Legacy

Incompatibilities:
- Deadlocks
- Livelocks
- SLA violations

Matching:
- Signature (WSDL)
- Behaviour (BPEL)
- QoS (WS-Security,...)

New functionality
Motivation: Adaptation in SOA

Alternative

- **flickr**

Legacy

- **Picasa**

Incompatibilities:

- Deadlocks
- Livelocks
- SLA violations

Matching:

- Signature (WSDL)
- **Behaviour** (BPEL)
- QoS (WS-Security,...)

New functionality

Google maps
Motivation: Adaptation in SOA

**Alternative**

- *flickr*™
- *Google maps*

**Legacy**

- *Picasa*™

**Incompatibilities:**
- **Deadlocks**
- **Livelocks**
- **SLA violations**

**New functionality**

**Matching:**
- Signature (WSDL)
- **Behaviour** (BPEL)
- QoS (WS-Security, ...)

**adaptor**

Contract
Motivation: Adaptation in SOA

Alternative

flickr™

Incompatibilities:
- Deadlocks
- Livelocks
- SLA violations

New functionality

Google maps

Legacy

Picasa™

Matching:
- Signature (WSDL)
- Behaviour (BPEL)
- QoS (WS-Security,...)

Applications:
- Discovery of adaptable WSs
- Mashups
- Orchestration or Choreography
- Dynamic (context-aware) adapt.
- Security adaptation
Moving to the IoT

**Alternative**

- **Incompatibilities:**
  - Deadlocks
  - Livelocks
  - SLA violations

**Legacy**

- **Matching:**
  - Signature (WSDL)
  - **Behaviour** (?)
  - QoS (WS-Security,...)

**New functionality**

Applications:
- Discovery of adaptable WSs
- Mashups
- Orchestration or Choreography
- Dynamic (context-aware) adapt.
- Security adaptation

---

**Google maps**
Moving to the IoT

Alternative

Incompatibilities:
- **Deadlocks**
- **Livelocks**
- **SLA violations**

New functionality

Legacy

Matching:
- **Signature** *(WSDL)*
- **Behaviour** *(?)*
- **QoS** *(WS-Security,...)*

Applications:
- Discovery of adaptable WSs
- Mashups
- Orchestration or Choreography
- Dynamic (context-aware) adapt.
- Security adaptation

DPWS

Google maps
Moving to the IoT

Alternative

Incompatibilities:
- Deadlocks
- Livelocks
- SLA violations

New functionality

Legacy

Matching:
- Signature (WSDL)
- Behaviour (?)
- QoS (WS-Security,...)

Applications:
- Discovery of adaptable WSs
- Mashups
- Orchestration or Choreography
- Dynamic (context-aware) adapt.
- Security adaptation

Google maps
1. Extend DPWS to include behavioural information in the WSDL as constrains (this work, [1, 2])
2. Synthesise behavioural adaptors for restricted devices [3]
3. Verify (adapted) compositions of things (future work)

What can be adapted?

- Cases:
  - Message names,
  - Extra argument which is not needed,
  - An extra optional branch,
  - Some "superfluous" messages,
  - Data and message reordering,
  - Message splitting and merging,
  - Data reutilization

\[
\langle a:!\text{request}(S,T); b:?\text{command}(S) \rangle
\]
What can be adapted?

**Cases:**
- Message names,
- extra argument which is not needed,
- an extra optional branch,
- some "superfluous" messages,
- data and message reordering,
- message splitting and merging,
- data reutilization.
What can be adapted?

- Cases:
  - Message names,
  - extra argument which is not needed,
  - an extra optional branch,
  - some "superfluous" messages,
  - data and message reordering,
  - message splitting and merging,
  - data reutilization.

```
<a:!request();b:?request()>
  <b:!ack()>
```
What can be adapted?

- Cases:
  - Message names,
  - extra argument which is not needed,
  - an extra optional branch,
  - some "superfluous" messages,
  - data and message reordering,
  - message splitting and merging,
  - data reutilization.

```
<b:!b()>
<a:!a(X,Y);b:?a(Y,X)>
<a:?b()>
```
What can be adapted?

Cases:
- Message names,
- extra argument which is not needed,
- an extra optional branch,
- some "superfluous" messages,
- data and message reordering,
- message splitting and merging,
- data reutilization.

```
<a:!login(P)>
<a:!request(R);b:?request(R,P)>
<a:?reply(D);b:!reply(D)>
```