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

WS4D Workshop
Rostock, Germany
29th November
2012

Summing-up

- Internet is evolving to the vision of IoT
- IoT could benefit from today’s WS architecture like Web does
- We like SOA approach (flexible, scalable, secure, interoperable)
- Goal: provide the functionality of each thing as a WS (TaaS), considering behaviour of things and verifying orchestration
- After studying several platforms, we chose Device Profile por Web Services (DPWS)
- With DPWS, two types of services: hosting and hosted services
Challenges

Services are described in **WSDL** and invoked using the normal **WS invocation**, however, only static interfaces are provided by most legacy discovery protocols.

- behaviour is required to specify the order of operations

Our contribution is to specify the behaviour of things and to verify the behavioural correctness of their compositions.

How? Adding **constraints** or **full-sequence FSM** to DPWS
- Facilitate the implementation of DPWS-compliant things
- Control the behaviour in dynamic environments

Motivating Example

**Example.** Airport surveillance system.

![Diagram of an airport surveillance system](image)
Motivating Example

November 29, 2012

Our proposal

DPWS

on, play, pause, stop, rewind, fast-forward, off
Our Approach: Constraints

We propose three types of constraints to specify the partial order in which the dependencies between messages are specified:

\[
\begin{align*}
\{ b_1, \ldots, b_n \} & \text{ afterAll } \{ a_1, \ldots, a_m \} \\
\{ b_1, \ldots, b_n \} & \text{ afterSome } \{ a_1, \ldots, a_m \} \\
\text{onlyOneOf } & \{ a_1, \ldots, a_m \}
\end{align*}
\]

Example. Constraints

\[ C_{A1} : \{ \text{move, record, zoom} \} \text{ afterAll } \{ \text{auth} \} \]
\[ C_{A2} : \{ \text{halt} \} \text{ afterAll } \{ \text{record} \} \]

We include the behaviour in the <documentation> tag in WSDL with the tag <behaviour>.

Our Approach: Finite State Machines

When it is not enough by specifying the partial order, but also the states changes according to the messages execution (complex relations), we propose to use Finite State Machines.

Example. FSMs
Our tool

It makes possible design and define both the constraints or the state machines in a graphical way.

---

Demo

Short demo:
- Run our example in WS4D-JMEDS tool.
- Discover devices using WS4D-Explorer
- Open WSDL in our GUI and play with it
- Save new WSDL with constraints / FSM tags
Our Approach:
Verification of behaviour-aware mashups of things

To specify composition of things, we propose a process model of a mashup of things represented as a graph; and we verify the correctness of the orchestration.

Example. FSMs

Conclusions and benefits

- Simple and concrete notation for representing the behaviour of things based on constraints and FSMs.

- Foundation to enable the creation of complex mashups by means of the composition and integration of services and heterogeneous objects.

- Mashups facilitate the exponential building of new more complex and sophisticated services (as SaaS or MaaS)
Future Work

- Keep working on the GUI
- Compose behaviour-aware things by extending any existing things-oriented mashup platform (WS4D-Pipesbox?)
- Generate mashups considering semantic description, and context and social information
- Reusing of mashups by developing both dynamic context-aware adaptation and reconfiguration mechanisms

Related Work

- REST vs WS-* technologies [GuinardEtAlMobiQuitous11, PautassoEtAlWWW08]
- DPWS – WS4D team - FP7 European projects SIRENA, SODA, SOCRADES [BohnGolatowski-ICN-ICONS-MCL06]
- FET European project CONNECT drops interoperability barriers by synthesizing on the fly connectors [InverardiEtAlISET11]
- Service-oriented solutions for Home Network System or Smart Home [NakamuraEtAlSCC11], [ParraEtAlIJSM09]
- Mashups of heterogeneous things in WoT [GuinardTrifaWWW09, ParaimpuWWW12]
- ThingML - modeling language to efficiently provide communicating services on resource-constrained devices [FleuryEtAlMODELS11]
Thank you for your attention!

Javier Cubo, Antonio Brogi, Ernesto Pimentel, Félix Prado, J. Antonio Martín

{cubo,ernesto, felixprado, jamartin}@lcc.uma.es
http://www.lcc.uma.es/~{cubo,ernesto}
brogi@di.unipi.it
http://www.di.unipi.it/~brogi