The Missing Layer - Virtualizing Smart Spaces

Marc-Oliver Pahl and Georg Carle

pahl@net.in.tum.de   carle@net.in.tum.de

Technische Universität München
Wouldn’t it be cool to have crowd-sourced development for smart spaces too?

Lots of developers produce lots of applications

Developer investment in Android, iOS, WP7

<table>
<thead>
<tr>
<th>Platform</th>
<th>Publishers</th>
<th>Average apps/user</th>
<th>Mindshare 6 (％ of developers using the platform)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>70,000</td>
<td>4.1</td>
<td>67%</td>
</tr>
<tr>
<td>iOS (iPhone)</td>
<td>101,000</td>
<td>4.1</td>
<td>59%</td>
</tr>
<tr>
<td>iOS (iPad)</td>
<td>36,000</td>
<td>3.6</td>
<td>N/A</td>
</tr>
<tr>
<td>Windows Phone 7</td>
<td>7,000</td>
<td>3.6</td>
<td>36%</td>
</tr>
</tbody>
</table>

*Source: vendor releases, Gartner

Lots of developers produce lots of applications

Marc-Oliver@Pahl.de | The Missing Layer - Virtualizing Smart Spaces
1. Homogeneous API
2. Portable Applications
3. Dynamic Extensibility
4. Simple Development of Services
Research questions?

1. How can a homogeneous API to heterogeneous devices in smart spaces look like?

2. How can the portability (instance-comprehensive use) of applications be supported?

3. How can dynamic extensibility be realized?

4. How can the development of applications for smart spaces be supported to facilitate it?
What kind of support for software orchestration do we have today?

- abstraction via ontologies
- supported by gateways
- centralized orchestration

---


<table>
<thead>
<tr>
<th></th>
<th>centralized orchestration</th>
<th>supported by gateways</th>
<th>abstraction via ontologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Homogeneity</td>
<td>-</td>
<td>o</td>
<td>+</td>
</tr>
<tr>
<td>App Portability</td>
<td>--</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Dyn. Extensibility</td>
<td>-</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Service Simplicity</td>
<td>-</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

Control-, Gateway-
Which could be a good abstraction?

State!

<\(L^*, \text{luminance} < 1000 \text{Lx}, \text{light\_on}\)>
Introducing Virtual State

Services/Applications

Virtual State Layer

Knowledge Agent

Orchestration Logic

Advanced Reasoning

Gateways

Gateway Service

Sensors/Actuators

Physical World

It is DAY

Knowledge Brokerage

State Virtualization

5000Lx

GW

0x42

S

S

5000Lx

Knowledge Brokerage

Knowledge Tree

Advanced Reasoning

5000Lx

Tuple Space

E.g.: Linda, JavaSpaces, TSpaces, ...

Producer

Consumer

queries

subscribes

<String address, String typeID, String value>

<String address, typeID, String>
Virtual State Layer Tuple Space

- Full spatial and temporal decoupling
- Data-type-based search
- Unified interface: search, get/set, subscribe

- Hierarchically structured tuples
  - Semantical locator/ID split via type-based search
  - Multi-inheritance
  - Shared global data-type repository for convergence (Model Store)
  - Dynamically extensible data model

- Address-based subscriptions
<table>
<thead>
<tr>
<th>Feature</th>
<th>Centralized orchestration</th>
<th>Supported by gateways</th>
<th>Abstraction via ontologies</th>
<th>Virtual State Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Homogeneity</td>
<td>-</td>
<td>o</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>App Portability</td>
<td>--</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Dyn. Extensibility</td>
<td>-</td>
<td>o</td>
<td>o</td>
<td>++</td>
</tr>
<tr>
<td>Service Simplicity</td>
<td>-</td>
<td>o (+/0)</td>
<td>o (+/0)</td>
<td>+ (+/)</td>
</tr>
</tbody>
</table>

- Centralized orchestration
- Supported by gateways
- Abstraction via ontologies
- Virtual State Layer

**Dyn. Extensibility** and **Service Simplicity** both show support (+) for abstraction via ontologies (+/0).
How do orchestration tasks look like?

$L^*\text{-NotificationHandler:}$
If ($\text{luminance < 1000Lx}$) then
light = on;

$L^*, \text{luminance < 1000Lx}, \text{light\_on()}>$
We and have a smart space with the VSL

WE DEVELOP software
The **Virtual State Layer** is part of the Distributed Smart Space Orchestration System (DS2OS)

- Provides autonomous knowledge storage and brokerage
- Written in Java
- Simple API: get/ set & publish/ subscribe
- Can be used via TCP socket, (OSGI, XMPP)
Crowd-Sourcing
Smart Spaces today?

Virtual State Layer

http://ds2os.org/

Thank you for your attention!
Questions?

Marc-Oliver Pahl and Georg Carle

Technische Universität München

The Missing Layer - Virtualizing Smart Spaces