Platform and language independent service life cycle management for device centric SOAs

Christian Fabian, Elmar Zeeb, Frank Golatowski, Dirk Timmermann
Outline

1. Motivation and Basics
2. Lightweight life cycle management
3. Installing process
4. Example of Use
5. Conclusion / Outlook
PDA

Mobile user

GPS

Sensor networks and observing of physical phenomena

Headset

telecommunication

Home automation ("the intelligent home")

Industrial automation

Automotive industry

Assembly line

Pervasive Computing

Multimedia Audio/Video

Health, Assisted Living

And many more …

Telecommunication

Office

Health, Assisted Living

Multimedia Audio/Video

Robots

Sensor networks and observing of physical phenomena

Home automation ("the intelligent home")

Industrial automation

Assembly line

Pervasive Computing

Multimedia Audio/Video

Health, Assisted Living

And many more …
# Devices Profile for Web Services – in detail

- message exchange (SOAP)
- Metadata device and service description (WS-MetadataExchange)
- handling large data as attachment (MTOM)
- Device description (WSDL)
- Data format and data type (XML Schema)
- Publish/Subscribe mechanism (WS-Eventing)

<table>
<thead>
<tr>
<th>Application-specific protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-Discovery</td>
</tr>
<tr>
<td>WS-Eventing</td>
</tr>
<tr>
<td>WS-MetadataExchange</td>
</tr>
<tr>
<td>WS-Transfer</td>
</tr>
<tr>
<td>Security</td>
</tr>
<tr>
<td>WS-Policy</td>
</tr>
<tr>
<td>WS-Addressing</td>
</tr>
<tr>
<td>SOAP</td>
</tr>
<tr>
<td>MTOM</td>
</tr>
<tr>
<td>WSDL</td>
</tr>
<tr>
<td>XML Schema</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
</tr>
<tr>
<td>HTTP</td>
</tr>
<tr>
<td>TCP</td>
</tr>
<tr>
<td>IPv4/IPv6/IP Multicast</td>
</tr>
</tbody>
</table>
Motivation

- Using the DPWS toolkit from the University of Rostock
  - Developed for embedded systems
  - available under GPL/LGPL license
  - Ad-hoc device discovery
  - WS-Eventing
  - Based on well known protocols and WS-specifications
  - Working with Windows Vista / 7 / .net Micro
Advertising device description with WS-Discovery to network.

DPWS device

Device

Hosting Service

Hosted Service

Life Cycle Manager

Service under LCM administration

Monitor

Metadata Repository

Put Metadata

Advertising

Start / Stop
Installing a new service

- ServiceID is used to distinguish several installed services
- Bundle content:
  - Service: executable file
  - WSDL: interface description of the service
  - XML: metadata description
  - Scripts (resolve-, start-, stop- and isAlive script)
  - Further files (e.g. Library, pictures, ... )
LUA for independent control of services

- Broker to translate between the life cycle manager and services
- Small sized interpreter (120kB)
- Lua to start/stop services language and platform independent
- Read environment variable via "os.getenv"
- Start service via "os.execute"

```lua
1  | -- test the operating system
2  | if os.getenv("OS") == "Windows NT" then
3  |     print("Windows") ;
4  |     if Resolve() == 0 then
5  |         -- start the service
6  |         os.execute("./simpleExample.exe")
7  |     else
8  |     print("Resolve not successful!")
9  |     return 0
10 | end
11 | else
12 |     print("OS is not Windows!")
13 |     return 0
14 | end
15 | return 1
```
## Example of Use on the FOX-Board LX832

### SOFTWARE features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux Kernel</td>
<td>Up to version 2.6.19</td>
</tr>
<tr>
<td>Server</td>
<td>HTTP (Web), FTP, SSH, Telnet</td>
</tr>
<tr>
<td>Language</td>
<td>C, C++, PHP, Python, etc.</td>
</tr>
</tbody>
</table>

### HARDWARE features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>66 x 72 mm (2.6 x 2.8 inches)</td>
</tr>
<tr>
<td>CPU</td>
<td>32bit RISC, 100 Mhz</td>
</tr>
<tr>
<td>Memory</td>
<td>8MB FLASH, 32 MB RAM</td>
</tr>
<tr>
<td>Power</td>
<td>5V 280 mA (1W)</td>
</tr>
<tr>
<td>Ports</td>
<td>1 Ethernet (10/100 Mb/s)</td>
</tr>
<tr>
<td></td>
<td>2 USB 1.1</td>
</tr>
<tr>
<td></td>
<td>1 serial console port</td>
</tr>
</tbody>
</table>
Conclusion

✔ Service-based life cycle manager for an implementation of DPWS

✔ Manage the services of a device during runtime through remote reconfiguration

✔ Platform and language independent by using a script-based interaction

✔ Transfer and run the concept on an embedded system

✔ able to modify a device during runtime
Outlook

- Support more platforms
- Implementing security functionality
- Possibility to move a service between several devices
- Reducing memory consumption
Thank you for your attention