DEVICE INTEGRATION IN SODA USING THE DEVICE DESCRIPTION LANGUAGE

Chao Chen and Abdelsalam (Sumi) Helal
INTRODUCTION

- The Pervasive Internet ➔ a wild jungle of devices and gadgets.
- Heterogeneity of
  - Devices:
    - From pin-head sensors to complex devices & appliances
  - Networks:
    - Ethernet, WiFi, Bluetooth, ZigBee, …
  - Running Environments:
    - None, OSGi, Jini, UPnP…
  - Device standards:
    - SensorML, IEEE 1451, ECHONET, Device Kit…
THE HARD-WIRING PROBLEM

- The typical ad-hoc strategy to integrate a device
  - A steep learning curve and laborious hacking experience
    - Needs to examine its interface and study the device protocol to establish a connection.
    - May not produce re-usable code or know-how.
  - The lack of standard on both hardware and software ends
    - No automatic way to link the numerous and heterogeneous device I/O
    - Programmers have to manually write code to associate devices with specific tasks.
- The problem becomes even more acute when
  - the number of device types continues to grow → the number of interfaces, connections, protocols multiplies;
  - One considers how systems need to evolve throughout their life-cycles.
OUR RESPONSE AND APPROACH

- Automate the process of device integration, so it would require **0-effort** from system integrators and pervasive system builders.
- Create an integration technology and associated proposed open standard (SODA) to allow device and sensor vendors to introduce such self-integrating products into the market place.
  - DDL - A descriptive language that describes a broad range of devices.
  - ATLAS - a service-oriented reference architecture for device integrations.
  - DDL / ATLAS - A proposed implementation of the SODA standard.
Talk Outline

- ATLAS: a service-oriented reference architecture for sensor and device integration
- DDL: Device Description Language - schema and its reference implementation
- Case Study: constructing a self-sensing space in the Gator Tech Smart House (GTSH)
- Standardization through SODA
- Related work
- Conclusion
ATLAS: A Reference Architecture for Service Oriented Sensor Platforms
THE ATLAS ARCHITECTURE

Application Layer
IDE  Service Composer  Context Builder

Services Layer
Query Processing Engine  Phenomenon Detection & Tracking Manager  Virtual Sensor Manager  Device Service
Atlas Middleware  Configuration Manager  Web Interface  Bundle Repository  DDL Bundle Generator

Node Layer
Atlas Sensor Platform  Sensor Ref  Actuator Ref  Complex Device Ref

Physical Layer
Sensors  DDL descriptor
Actuators  Complex Devices  DDL descriptor

Internet

OSGi Framework
ATLAS PLATFORM
Device Description Language (DDL)
Language Schema and its Implementation
MODELING DEVICES IN DDL

A DDL device model

DDL classifies devices into 3 categories:

- Sensors
- Actuators
- Complex Devices
THE DDL LANGUAGE SCHEMA

- DDL uses XML encodings.
- Readable to both human and machine.
- A DDL schema defines the constraints on the structure and the content of a DDL document.
- The schema will be enforced by the DDL validity checker, a component of the DDL language processor.

```xml
<xsd:complexType name="Device">
  <xsd:sequence>
    <xsd:element name="Description" type="DescriptionType" minOccurs = "1" maxOccurs="1" />
    <xsd:element name="Interface" type="InterfaceType" minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
```
What’s Inside a DDL Descriptor File?

- Each DDL descriptor file describes a single type of device.
- It contains:
  - Information for service registration and discovery
    - e.g., device name, model, function description, etc.
  - Description of device operations
    - each operation is a collection of input/processing/output function chains
    - the low-level communication between a device and its service are represented as ‘Signals’
    - the high level semantics of signals are ‘Readings’
AN EXAMPLE:
TMP36 ANALOG TEMPERATURE SENSOR

- An analog sensor:
  - “Signal”: a converted value output from the ADC port on the sensor platform.
  - “Reading”: the temperature value in centigrade.
- DDL defines
  - the semantics of a temperature reading
  - the process of the **signal to reading** conversion
A complex device:

- **“Signal”**: the byte stream output from the serial port.
- **“Reading”**: the blood pressure measurement converted from the byte stream.
Role Implications (without ATLAS/DDL)

The system integrator is the sole player in the field and has to deal with both hardware and software complications.
Role Implications (with ATLAS/DDL)

- Multiple roles are engaged;
- Their responsibilities are clearly separated by both hardware and software abstractions.
Case Study: Constructing a Self-Sensing Space in the Gator Tech Smart House
CASE STUDY

Self-Sensing Space:
- An intelligent environment that recognizes its devices and services, interpret their status, and generate a model of the space.

The challenges:
- Integration of dumb objects: a self-sensing space should not ignore everyday objects such as furniture and electric appliances.
- End-to-end self-integration: both smart devices and dumb objects should be seamlessly self-integrated into the space.
APPLICATION SCENARIO

Remote Caregiver

Smart Devices
Dumb Objects

Atlas Platform

Smart Plug

Home Network

Atlas Middleware

Service Bundle
Service Bundle

Internet

Self-Sensing 3D Interactive Model

Remote Caregiver

Interaction

Device Bundle Factory

System Integrator

DDL Device Descriptor

Smart Plug

Smart Devices
Dumb Objects
Standardization through SODA
STANDARDIZATION

- DDL is a proposed implementation of the Service-Oriented Device Architecture (SODA) standard framework.
- SODA is
  - an emerging standard alliance,
  - an extension to SOA to incorporate devices in distributed enterprise systems.
- When modeled as a service, device access and control can be made available to a wide range of enterprise applications using SOA mechanisms.
The DDL language specification and its software are available online at

http://www.icta.ufl.edu/atlas/ddl/
Related Work
There have been a number of standards proposed:

- **ECHONET**
  - The Energy Conservation and Homecare Network standard, initiated in Japan.

- **IEEE 1451**
  - The IEEE standard for smart transducer interfaces.

- **SensorML**
  - The Sensor Model Language, initiated in the geospatial community.

- **Device Kit**
  - An IBM implementation of the SODA architecture
## Comparison of Standards

<table>
<thead>
<tr>
<th>Key Comparisons</th>
<th>ECHONET</th>
<th>IEEE 1451</th>
<th>SensorML</th>
<th>Device Kit</th>
<th>DDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>Class specification in plaintext</td>
<td>Interface Definition Language</td>
<td>XML</td>
<td>XML</td>
<td>XML</td>
</tr>
<tr>
<td>Design perspective</td>
<td>Object-oriented</td>
<td>Modular</td>
<td>Data-oriented</td>
<td>Modular</td>
<td>Data-oriented</td>
</tr>
<tr>
<td>Device Model</td>
<td>Single object</td>
<td>Multiple blocks</td>
<td>Process chain</td>
<td>Multiple layers</td>
<td>Single device, cross layer</td>
</tr>
</tbody>
</table>

### Other Comparisons

<table>
<thead>
<tr>
<th>Basic component</th>
<th>ECHONET</th>
<th>IEEE 1451</th>
<th>SensorML</th>
<th>Device Kit</th>
<th>DDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Device</td>
<td>Block</td>
<td>Process</td>
<td>Device function layer</td>
<td>Device</td>
</tr>
<tr>
<td>Composite component</td>
<td>NA</td>
<td>Device</td>
<td>Process Chain</td>
<td>Device</td>
<td>Derived virtual sensor</td>
</tr>
<tr>
<td>Measurement modeling</td>
<td>Primitive data types</td>
<td>Complex data types</td>
<td>Complex data types</td>
<td>Primitive data types</td>
<td>Primitive data types with aggregation</td>
</tr>
<tr>
<td>Protocol modeling</td>
<td>Inexplicit</td>
<td>Explicit</td>
<td>Inexplicit</td>
<td>Explicit</td>
<td>Explicit</td>
</tr>
<tr>
<td>Software support</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>DKML parser and Eclipse plug-in</td>
<td>DDL Language processor</td>
</tr>
<tr>
<td>Specification</td>
<td>Published</td>
<td>Published</td>
<td>Published</td>
<td>Only schema available</td>
<td>Published online</td>
</tr>
</tbody>
</table>
Conclusion
CONCLUSION

- The scale and pace by which the Pervasive Internet is evolving today demand a new breed of integration technology that is scalable and automatic.

- The Device Description Language within the Atlas sensor platform and middleware is capable for describing and integrating a great variety of devices ranging from a pinhead sensor to a complex device. Currently sensors and complex device integration are supported.

- We are now working on an improved design of the Atlas firmware to better support actuator integrations.
Thank you!