Digital Object Memories for the Web of Things

Dr.-Ing. Jens Haupert
German Research Center for Artificial Intelligence (DFKI)

Web of Things Workshop, June 25th 2013
Sample Scenario

- **Linking:** Physical Object and Object Memory
- **Monitoring:**
  - Temperature
  - Humidity
  - Acceleration

* Recycling

* Transport Data
* Usage

* Order
* Components
* Quality Checks

* Price
* Date of Sale
* Storage Conditions
Open-Loop Scenarios

Closed-Loop

Open-Loop
Open-Loop Scenarios (2)

- A large set of stakeholders
- Ad-hoc lifecycle chain
- No a priory definitions
- Heterogeneous datasets
New Requirements

• Life Cycle Chain
  – Object-related information
  – Open-loop distribution

• Maintenance
  – Specific configurations / Lot size one
  – Retrieving spare parts

• After Sales
  – (Remote) condition-based monitoring

• Improved Recycling
  – Provide information about components and elements
  – Recycling feedback to manufacturer
Data Foundation

OMM → OMS → Processing → Activity

Data Foundation → Infrastructure
Object Memory Model

- *Structure model* to partition heterogeneous object memory data
- Group associated data to *blocks*
- Each block consists of actual *payload* and additional *meta data*
- Payload can be out-sources to *external media*
- Abstract model with XML- and embedded HTML5-representations
- W3C® Incubator Group (OMM-XG)
Object Memory Model (2)
Object Memory Model (3)

Block Metadata

Block Payload

ID

Namespace

Format

Type

Subject

Creator

Contributors

Title

Description

Content Description

M2M

Access History

Clear Text Data

HCI
Object Memory Model (4)

Block 1  Block 2  Block 3  ...  Block n

Stakeholder A  Stakeholder B  Stakeholder Z
Defined Blocks

- OMM Structure Block
- OMM IDs-Block
- OMM Key-Value-Template
- OMM+ Semantics Block
- OMM+ Embedded Block
- OMM+ PiVis Block
- OMM-XG
- Extensions
Sample Memory (1)

OMM-Block with sensor readings

<table>
<thead>
<tr>
<th>ID</th>
<th>block_123</th>
</tr>
</thead>
</table>
| Format      | application/xml  
  Schema: http://mycomp.com/sensor.xsd |
| Creator     | 123456789@DUNS, 16.01.2013 16:15 |
| Title       | “SensorReading (temp/humi/shock)” |
| Type        | http://purl.org/dc/dcmitype/Dataset |
| Subject     | “sensorReadings”  
  http://mycomp.com/o.owl#Temperature  
  http://mycomp.com/o.owl#Humidity  
  http://mycomp.com/o.owl#Shock |

```xml
<?xml version="1.0">
<log>
  <reading>
    <value>-6</value>
    <topic>#Temp</topic>
    <unit>#Celsius</unit>
  </reading>
  <reading>
    <value>80</value>
    <topic>#Humid</topic>
    <unit>#Percent</unit>
  </reading>
  <reading>
    <value>1.6</value>
    <topic>#Shock</topic>
    <unit>#G</unit>
  </reading>
</log>
```
Sample Memory (2)

OMM-block with embedded memory

Integration
## Sample Memory (3)

### OMM-Block with embedded memory

<table>
<thead>
<tr>
<th>ID</th>
<th>block_456</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>urn:ommplus:blocks:embeddedBlock</td>
</tr>
<tr>
<td>Format</td>
<td>application/xml</td>
</tr>
<tr>
<td>Creator</td>
<td>123456789@DUNS, 16.01.2013 16:15</td>
</tr>
<tr>
<td>Title</td>
<td>“Memory: Sensor Optobero 124”</td>
</tr>
<tr>
<td>Type</td>
<td><a href="http://purl.org/dc/dcmitype/PhysicalObject">http://purl.org/dc/dcmitype/PhysicalObject</a></td>
</tr>
<tr>
<td>Subject</td>
<td>„primaryID.opto124“ „myLoggingInfo“ <a href="http://ont.org/o.owl#Maintenance">http://ont.org/o.owl#Maintenance</a></td>
</tr>
</tbody>
</table>
Infrastructure

- Data Foundation
- Infrastructure
- Processing
- OMM
- OMS
- Activity
Object Memory Server

• Modular Architecture
  – Complete feature set with dedicated server
  – Partial functionality with embedded systems
  – Minimal core with simple systems
Object Memory Server (2)

RESTful Interface

- `<memory name>`
  - `toc` > GET
  - blockIDs > GET
  - block / `<block ID>` / creator > GET | PUT
    - subject > GET | PUT
    - type > GET | PUT
    - payload > GET | PUT
    - ...

Role-based Access

Version Control
Processing

Data Foundation

OMM

Infrastructure

OMS

Processing

Activity
Activity

• Previous Architecture:
  – Passive data storage only

• Add processing logic to object memories

• Execute small, local operations (snippets) within object memory
Activity (2)
Activity (2)

Object Memory

- Logic
- Knowledge
- Result

Activity Module

- Interpreter
- Code Execution in Sandbox
- Heartbeat
- REST Interface
- Memory Events

Cyber-physical System

User
Activity (4)

1. Memory URL
2. Installed Snippets
3. Snippet Store
4. Upload

Snippet Selection by User

Activity Modules
Memory
- Logic
- Knowledge
ADOMe vs. CPS

Digital Object Memory

Active DOMe (ADOMe)

Cyber-physical System

Memory

Knowledge

Activity Modules

Logic

Knowledge

Sensors / Actuators

Activity Modules

Logic

Knowledge
Object Memory Server (OMS)
- Java-Version w/o VCS und ACL
- RESTful interface
- Activity Module and Snippets
- OMM XML representation

Object Memory Server (OMS)
- C# compact implementation
- RESTful interface
- Transparent sensor integration
- OMM Binary representation
Visualization, Tools and Extensions

Data Foundation

OMM

Infrastructure

OMS

Processing

Activity
Database Migration

- *Today*: Data stored in databases
- *The day after tomorrow*: Digital object memories

- Migration:
  - Semantic Harmonization via Ontology Model
  - Facts and File-extraction
  - Automatic Memory generation
Data Representation

Pros:
- Standardized / Well defined
- Widely used
- Human readable

Cons:
- Verbosity and complexity
- Slow processing
- Bad cost-value ratio
Data Representation (2)

- **Dynamic mapper for binary representation**
  - Lossless conversion for low storage space consumption
  - Lossy conversion for minimum space consumption
    - Partial restore of lost data possible via mapping table
Secure Provenance

• Warranty deeds and legal grounds
  – Secure and verifiable data necessary

➢ Memory with secure provenance
Secure Provenance

- No later modification feasible
  - No alteration of existing blocks
  - No insertion of new blocks
  - No deletion of existing blocks (with Authority Server)
Results

• Definition of a **meta model** to structure heterogeneous object memory data and positioning in W3C OMM-XG

• *Reference implementation* of a OMM-based memory infrastructure for open-loop scenarios

• *Hardware abstraction* allows usage independent of object properties and capabilities

• Plug-In-based, dynamic *visualization* for heterogeneous data

• Additional *toolset* to create new applications and migrate existing systems
Discussion

• DOMe-related Data by Trumpf?

• Where could digital object memories be used?
  – Manufactured products?
  – Machinery?

• New potentials and business models?
  Possible risks?
Discussion

- Which types of memories are suitable?
  - Embedded/on-object
  - Server-based/off-object

- What data could enrich the product life-cycle?

- Where could autonomous activity be used?

- How could a Snippet Store look like?
Thank you very much for your attention.

http://www.dfki.de/omm-tools/
Benefits

• Open data storage
  – Communicate with open-loop partners
  – Standardized formats and proprietary extensions

• Self-explaining objects
  – Object-related information attached to physical object
  – Replace paper documentation
  – Define monitoring conditions
  – Maintenance data

• Maintenance data attached to product
  – Retrospective
  – Spare parts
Benefits

• Object memory data storage location
  – On-object for high value products
  – Off-object for low cost products
  – Arbitrary hybrid solutions possible
  – Communication via standardized web techniques

• Versatile applications for object memories
  – Raw materials
  – Machinery
  – Goods
  – Hierarchical solutions
Benefits

• Object memory as storage space for snippets

• Secure *sandbox* execution

• Update Snippets via AppStore
  – Subsequent parameter adjustment

• Object-related logic (lot size one)
  – Integrity Check

• Autonomous/independent monitoring
  – Independent Object Monitoring