When Devices Become Collaborative
Supporting Device Interoperability and Behaviour Reconfiguration Across Emergency Management Scenario

Mihaela Brut, Patrick Gatellier - Thales Services
Ismail Salhi, Sylvain Cherrier, Yacine Ghamri - Univ Paris Est
David Excoffier - Sogeti HighTech
Nicolas Dumont, Mario Lopez-Ramos - Thales Comm. & Security

7th of March 2014
Introduction & Context Scenario

State of the Art

- IoT approaches to device data and service management
- Service Composition Approaches
- Data Formats for Emergency Management Situations

WoO Solution for Dynamic Devices Collaboration

- Web of Objects approach in Device Data Management
- Services Workflow for Device Cooperation
- Data exchange Model for Incident Management

Conclusions and Further Work
Introduction & Context Scenario
Emergency management is a highly critical domain

- Multiple stakeholders with different missions and who have specific incident management procedures
- Have to share the same space and objects and to receive the suitable information in real time
- Real-time alarm processing leading to stakeholders coordination could save lives and resources

Example of different stakeholders mission in a public area

- Control center: manage and supervise all the detection equipments;
- Surveillance company: detect intruders and assure the security;
- Train/Air company: communicate with the passenger about his trips;
- Area Administration: administer and keep up the shared space
- Shops: to sell goods and make publicity for their business offer;
- Passengers: to travel, relax, enjoy himself, buy;
- First responders: re-establish the normality after an incident;
# Objects Acting in an Emergency Scenario

<table>
<thead>
<tr>
<th>Objects</th>
<th>Associated Services</th>
<th>Target Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door</td>
<td>Door_intrusion-detected;</td>
<td>CC-Alarm-Manager</td>
</tr>
<tr>
<td>Emergency door (ED)</td>
<td>ED_Open_Notification; ED_Closed_Notification;</td>
<td>CC-Alarm-Manager</td>
</tr>
<tr>
<td>PTZC</td>
<td>PTZC_movement_detected; PTZC_photo_sending</td>
<td>CC-Alarm-Manager; PTZC; EE</td>
</tr>
<tr>
<td>Electrical Equipment (EE)</td>
<td>EE_autoMonitoring</td>
<td>CC-Alarm-Manager; PTZC;</td>
</tr>
<tr>
<td>Fire detector, Smoke detector, Temperature meter;</td>
<td>D-Fire_sendAlarm, D-Smoke_sendAlarm, M-Temperature_sendAlarm</td>
<td>CC-Alarm-Manager;</td>
</tr>
<tr>
<td>CC-Alarm-Manager</td>
<td>CC-AM_sendFireAlarm CC-AM_sendComplexAlarm</td>
<td>Fire-Agent-Smartphone; Security-Agent-Smartphone</td>
</tr>
<tr>
<td></td>
<td>CC-AM_openEmergencyDoor CC-AM_closeEmergencyDoor</td>
<td>ED ED Publicity-display-monitor</td>
</tr>
<tr>
<td></td>
<td>CC-AM_Evacuation-message</td>
<td></td>
</tr>
<tr>
<td>CC-Video-Tracker</td>
<td>CC-VT-suspectSelection</td>
<td>Surveillance-Camera-XYZ</td>
</tr>
<tr>
<td></td>
<td>CC-VT-suspectLocalization</td>
<td>Security-Agent-Smartphone</td>
</tr>
<tr>
<td></td>
<td>CC-VT-closeEmergencyDoor</td>
<td>ED ED</td>
</tr>
<tr>
<td>CC-Access-Control</td>
<td>Open/close doors</td>
<td>ED</td>
</tr>
</tbody>
</table>
### Objects Acting in an Emergency Scenario

**Observation:** a service could be « offered » based on the object behavior, or it should be demanded by an external object (that « discovered », or that have « registered » in order to be notified about it’s status changing)

EE – PTZC => when an object feels itself damaged, ask the closest camera to turn and to record its scene

<table>
<thead>
<tr>
<th>Objects</th>
<th>Associated Services</th>
<th>Target Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surveillance-Camera-XYZ</td>
<td>SC-XYZ-videoStreaming</td>
<td>CC-Video-Tracker</td>
</tr>
<tr>
<td>Fire-Agent-Smartphone</td>
<td>Fire-Agent-Smartphone_Ack</td>
<td>CC-Alarm-Manager;</td>
</tr>
<tr>
<td>Security-Agent-Smartphone</td>
<td>Security-Agent-Smartphone_Ack</td>
<td>CC-Alarm-Manager;</td>
</tr>
<tr>
<td></td>
<td>Security-Agent-Smartphone_Success</td>
<td>CC-Alarm-Manager;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security-Agent-Smartphone;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Police</td>
</tr>
<tr>
<td>Water-Plug</td>
<td>Water-Plug_Location</td>
<td>Fire-Agent-Smartphone</td>
</tr>
<tr>
<td></td>
<td>Water-Plug_Unauthorized-usage-alarm</td>
<td>CC-Alarm-Manager;</td>
</tr>
<tr>
<td></td>
<td>Water-Plug_Distribution-ack</td>
<td>CC-Objects-Manager??</td>
</tr>
</tbody>
</table>
**Maintenance scenario**: devices (objects) to be first installed, dynamically configured and replaced in case of dysfunction:

- An IP address is assigned to the device, and it is registered in the network, together with its technical details and its localization.

**Public security scenario**: suspect person to be tracked further to a malicious action;

**Incident scenario** (e.g. fire) affecting the infrastructure and requiring first responders;

**Commercial scenario**: music band/VIP/publicity car to be tracked and accompanied by displaying publicity announces on the corresponding display monitors;
Scenario with objects, actions and business workflows

1. All the **objects** are installed and configured;
2. A **VIP** is walking in the commercial area and is tracked by the Shops Center by using **CC-Video-Tracker**;
3. The **Publicity Display monitors** around the VIP current location display a VIP presentation.

- A person (the Suspect) enters into a restricted area and is detected by the **Door** and by the **PTZ low quality camera**;
- The suspect damages an **electrical equipment (EE)**. The EE raise an alarm, while the **PTZ camera** turns to the EE scene.
- A fire is triggered further to the EE damage, and is detected by the **Fire detector, Smoke detector, Temperature meter**;
- Further to received alarms, the **Control Center** calls the **Fire agency** and send a broadcast alarm to **security agents** in the station area, calling two of them in the damaged restricted area.
- The **fire agent** and the **security agent** confirm the mission.
- The **Control Center** initiates the video tracking process, having priority over the Shops Center, whose access to the **CC-Video-Tracker** will be interrupted;
The security agent catch the suspect and immobilize him, while notifying the CC.

The Shop Center re-gains access to the CC-Video-Tracker in order to continue the VIP tracking.

When the fire agent arrives in the station, the closest Water plug to the fire place notifies the agent about its position.

During the water distribution, the water plug notifies about its action the CC, the fire agent (and eventually the security agents).

Starting with the moment when the security alarm is validated, the Publicity Display monitors around the fire are used for displaying the security messages.

When the repairman arrives in the station, the location and technical details about the Electrical Equipment will be transmitted on his SmartPhone.
Camera: photo-based detection
Next camera: according location
Storage: further to alarm

Agent Smartphone: suspect photo and current location

Publicity panel: Evacuation message
Fireman Smartphone: in-door guiding
Hydrant: notifying its location and status

RoomB8 – Suspect & fire detected

Intrusion: Hall-Effect Sensor & PTZ camera
Fire: ambient sensors
EE damaged: electricity sensor
Suspect photo: PTZ camera

CC-Video-Tracker:
- suspectSelection
- suspectLocalization

CC-Alarm-Manager:
- callFireAgency
- assignRightsFireman
- evacuationMessage

CC-Maintenance:
- equipmentFaultDetection
- assignRightsRepairman

Repairman Smartphone:
- in-door guiding
- technical details

Repairman guided and informed

Fireman & people coordinated

Control Center

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT
## The Owners and the Business Workflows of the Objects

<table>
<thead>
<tr>
<th>Objects</th>
<th>Owners</th>
<th>Business Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door</td>
<td>Area Administration (e.g. Airport, Shopping Mall)</td>
<td>Maintenance, Incident</td>
</tr>
<tr>
<td>Emergency door (ED)</td>
<td>Area Administration</td>
<td>Maintenance, Incident</td>
</tr>
<tr>
<td>PTZC</td>
<td>Surveillance company</td>
<td>Public security, Commercial</td>
</tr>
<tr>
<td>Electrical Equipment (EE)</td>
<td>Area Administration</td>
<td>Maintenance, Incident</td>
</tr>
<tr>
<td>Fire detector, Smoke detector, Temperature meter;</td>
<td>Area Administration, Control Center</td>
<td>Maintenance, Incident</td>
</tr>
<tr>
<td>CC-Alarm-Manager</td>
<td>Control Center</td>
<td></td>
</tr>
<tr>
<td>CC-Video-Tracker</td>
<td>Surveillance company</td>
<td></td>
</tr>
<tr>
<td>CC-Access-Control</td>
<td>Control Center</td>
<td></td>
</tr>
<tr>
<td>Surveillace-Camera-XYZ</td>
<td>Surveillance company</td>
<td></td>
</tr>
<tr>
<td>Fire-Agent-Smartphone</td>
<td>First Responders</td>
<td></td>
</tr>
<tr>
<td>Security-Agent-Smartphone</td>
<td>First Responders</td>
<td></td>
</tr>
<tr>
<td>Water-Plug</td>
<td>Area Administration</td>
<td></td>
</tr>
</tbody>
</table>
**Important issues of the scenario**

- Maintenance workflow: devices have to be configured and managed.
- All business workflows: devices’ services have to be executed in specific sequences.
- Messages transmitted between devices should automatically exploitable by all Emergency Management procedures.

**State of the Art**

- IoT approaches to device data and service management.
- Service Composition Approaches.
- Data Formats for Emergency Management Situations.

**WoO Solution for Dynamic Devices Collaboration**

- Web of Objects approach in Device Data Management.
- Services Workflow for Device Cooperation.
- Data exchange Model for Incident Management.
State of the Art & WoO Contribution

Device management
Device management: SCADA Traditional Approach

The Station Control Center communicate via an integrated protocol with each object:

- Transmit commands,
- modify certain object properties,
- ask for information.

Open/close the station! Update infos in real-time

I have 7 possible variants

Locate a man with red hat descending from the Paris train

5 minutes until London train 5 platforms distance

Line changed for Paris train

Go to line 8
### Device technologies in the Internet of things

#### IoT: Autonomy, adaptability, reactivity

<table>
<thead>
<tr>
<th>Schema</th>
<th>UPnP</th>
<th>DPWS</th>
<th>ONVIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addressing</td>
<td>DHCP, AutoIP</td>
<td>DHCP, AutoIP</td>
<td>WS-Addressing</td>
</tr>
<tr>
<td>Discovery</td>
<td>SSDP</td>
<td>WS-Discovery</td>
<td>WS-Discovery</td>
</tr>
<tr>
<td>Description</td>
<td>UDA Schema</td>
<td>WSDL</td>
<td>WS-I Basic Profile 2.0</td>
</tr>
<tr>
<td>Control</td>
<td>SOAP 0.9, 1.1</td>
<td>SOAP 1.2</td>
<td>ONVIF DM WSDL</td>
</tr>
<tr>
<td>Eventing</td>
<td>GENA</td>
<td>WS-Eventing</td>
<td>WS-Base Notification WS-Topics</td>
</tr>
<tr>
<td>Presentation</td>
<td>HTTP, HTML</td>
<td>HTTP, HTML</td>
<td></td>
</tr>
</tbody>
</table>

**Control (device management) means to implement:**
- software and firmware provisioning *(install, update, uninstall)*
- software control *(start/stop)*
- management *(get, set, create, delete)*
- diagnostics *(self-test, ping, tracert, nslookup)*

---

Probe: ticket (discovery approach)

ProbeMatch: ticket

My name is: iP@M
My properties are:
My services are:
operating context:
my journey

Event: 15 min to destination
Message: Wake up!

Control: GoTo Platform 8
Control Point: a client that discover and control the servers UPnP
Device: the server (that receive demands)
The same object could be CP and Device
Very popular in industry

Disadvantages:
- Require multiple resources at object level: Web server + xml parser + soap + ...
- Based on non-standard protocols: Discovery: SSDP on top of HTTPU, HTTPMU
  Description: dialect XML
  Events: GENA
- No authentication protocol (no security)
**Standard OASIS** since juin 2009,
Exclusively based on standards across all the layers:

<table>
<thead>
<tr>
<th>DPWS metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-Discovery</td>
</tr>
<tr>
<td>WS-Transfer</td>
</tr>
<tr>
<td>WS-MetadataExchange</td>
</tr>
<tr>
<td>WS-Eventing</td>
</tr>
<tr>
<td>WS-Addressing</td>
</tr>
<tr>
<td>SOAP 1.2</td>
</tr>
<tr>
<td>WSDL 1.1, XML Schema</td>
</tr>
<tr>
<td>UDP</td>
</tr>
<tr>
<td>HTTP 1.1</td>
</tr>
<tr>
<td>TCP</td>
</tr>
<tr>
<td>IPv4/IPv6</td>
</tr>
</tbody>
</table>

DPWS provides particular approaches (extensible) for:
- **Discovery**
  - Dynamic reconfiguration
- **Dynamic view of available devices**
- **Metadata exchange**
  - Semantic reasoning
- **Eventing**
  - Monitoring and reasoning

**Advantages:**
- The existing implementations interoperables

**Disadvantages:**
- Weak compatibility
**WoO solution**: management plugin for DPWS communication stack to allow device management through web services protocol

DPWS & management plugin can be used at every level: from IT down to devices:

- Forge website (open source code): https://forge.soa4d.org
- DPWS project: https://forge.soa4d.org/projects/dpwscore/
Main generic WS-Management plugin features:

- Device management with XML-based description of managed sensors/devices (a.k.a resources).
- Management Operations, using WS-* specifications:
  - DISCOVERY: Discover managed resources (WS-Discovery).
  - GET, PUT, CREATE, and DELETE: read, modify, create, or delete (portion of) resources (WS-Transfer). (Hey, looks like REST operations ;-)).
  - (UN)SUBSCRIBE: Subscribe/ Unsubscribe management events sent by resources through DPWS Heartbeat and also available (WS-Eventing).
WoO Interoperability: SHT smart engine – Big picture

Connect everything, everywhere, easily.
Adding specific local behavior to plugins

Specific local behavior, specific intelligence, can be added to any input/output plugins.

For example, adding a behavior to take into account data coming from different kind of sensors, and correlate them in real-time to output new higher-level events.

This behavior can be a simple embedded state chart, or a bigger intelligence artificial engine if needed.

This optional behavior, specific to each plug in, allow the implementation any kind of scenarios and business cases.
WoO – Business Workflows are employed to define choreographies.
State of the Art & WoO Contribution

Services Workflow for Device Cooperation
Orchestration vs composition

- Business Process Execution Language (WS-BPEL)
- Web services Choreography Description Language (WS-CDL)
**IoT Application Creation**

- Allows the design and execution of versatile IoT apps granting a wide panel of modular and incremental interactions for a wide variety of devices.
WoO Service Choreography approach

Dlite API for Your Devices

How does it work?

- **Java/Internet Capable Device**: Dlite API
  - **Door**: door intrusion detected
  - **PTZ**: movement detected
  - **Fire Detector**: send alarm

- **No Java/Internet Device**: Java/Internet capable Gateway

- **Java/Internet capable Gateway**:
  - **Fire AGT Phone**: confirm-mission
  - **Scryt AGT Phone**: confirm-mission
  - **Video Tracker**: suspect localization
  - **Maintenance**: assign repairman

Abstracted View of Devices on BEC3

ITEA2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT
2) D-LITE: Choreography
Message exchange in Incident Management
Current ontologies or vocabularies specialized in message exchanging between different stakeholders

- based on the e-mails and instant messages structure and functioning principles
- support for identifying the sender, the recipient(s), their address, the subject of a message, etc.

Emergency Data Exchange Language (EDXEL)

- standards family developed by OASIS
- follows the same principle, but adding support to describe:
  - the content of an alarm
  - the conditions under which an information control could be yield to an external authority such as the police.
EDXL-DE (Distribution Element)

- designed to package and deliver any OASIS EM Standard or other data message;
- may be thought as a “container” that carries a “payload” of formatted message sets (such as Alerts or Resource Messages);
- Messages may be distributed to specific recipients, to a geographic area, or to an agency type (police, fire, etc.).

EDXL-CAP (The Common Alerting Protocol)

- provides support for multiple alert messages types, such as:
  - Warnings, Acknowledgements, Expirations and cancellations, Updates and amendments;

EDXL-RM (Resource Messaging)

- describes a set of pre-defined messages for requesting/sharing resources
- (e.g. emergency equipment, supplies, people);
EDXL based message exchange model

```
MessageModel
+ createMessage() :void
+ sendMessage() :void

Header
- subject :char
- type :char
- dateTimeSent :int
- language :char
- keywords :char
- confidentialityLevel :int
- deliveryStatus :int
- ID :int
- sender :Person
- recipient :Person
- source :Person
- targetArea :Area
- targetAudience :char

Content
+ defineAlert() :void
+ defineResource() :void
+ exchangeResource() :void

Alert
- category :char
- concernedResource :Resource
- certainty :int
- instruction :char
- severity :int
- urgency :int
- ID :char
- parameter :char
- value :int
- recommendedAction :char
- contact :Person

Resource
- credentials :char
- description :char
- ID :char
- name :char
- responsibleParty :Person
- specialRequirements :char

Area
- longitude :int
- altitude :int
- radius :int
- description :char

ResourceExchange
- exchangedResource :Resource
- exchangeType :char
- arrivalRequested :int
- arrivalEstimated :int
- arrivalActual :int
- departureRequested :int
- departureEstimated :int
- departureActual :int
- cost :int
- funding :char

Person
- name :char
- phone :int
- e-mail :char
- fax :int
- address :char
- role :char

File
- mimeType :char
- size :int
- URL :char
```
Conclusions
Semantics is a key enabler for a common language such as:

- A device (belonging to a particular or to a institution) is accessible and to third parties applications (government, police, fire agency, hospitals, shops, restaurants)
- A complex application for smart city integrate as actors devices from multiple stakeholders

Further work

- Develop a semantic alternative of the BEC3 approach, where the composition is based on ontology

Further Steps towards finalizing the WoO solution:

- Standardization of the proposed framework for the IoT
- Validating the solution through all the project demonstrators

Further collaboration perspective

- object virtualization : integrating semantic approach with the cloud computing technologies