Middleware Support for Generic Actuation in the Internet of Mobile Things

Sheriton Valim
Matheus Zeitune
Bruno Olivieri
Markus Endler

Laboratory for Advanced Collaboration (LAC)
Pontifícia Universidade Católica of Rio de Janeiro (PUC-Rio)
Brasil
Agenda

1. Introduction
2. Challenges
3. Generic Actuation
4. Our Approach
5. M-Act
6. SOM
7. Driver
8. Tests
9. Conclusions and Future Work
Introduction

- Actuators are essential to IoT Applications
  - Smart Homes
  - Smart Industry
  - Smart Transportation
  - Healthcare

- Change their physical environment
Challenges

- Process and memory limitations
- Short-Range Communication
- Need gateway to connect to the Internet
Challenges

- Diversity
  - Different devices
  - Different implementations
  - Different command protocol
Generic Actuation

- What?
  - Control any actuator
  - Independent protocol
  - Independent manufacturer

- How???
Our Approach

- Taxonomy of Actuation
  - Blind
  - Reliable with active feedback
  - Reliable with passive feedback
  - Noticeable with indirect feedback
  - Noticeable with indirect and delayed feedback
Our Approach

- **ContextNet Extension**
- **M-ACT**
  - Microservice of M-Hub
  - Request drivers for the actuators
  - Translates generic commands to native protocol of actuators
- **SOM (Smart Objects Manager)**
  - Microservice of the ContextNet Core
  - Repository of drivers
  - List of devices registered
  - ConnectedTable
- **Drivers to describe the native protocol**
Our Approach
Our Approach

- MACTQuery

```
{  
  "MACTQuery" :  
  {  
    "type" : "cmd|driver",
    "label" : "command_label",
    "target" : "mobject id|mobject group id|...",
    "cmds" : 
    [  
      {  
        "seq" : 0,
        "cmd" : "move|setColor|...",
        "args" : "cmd arguments accordingly to driver description"
      },  
      {  
        "seq" : 1,
        "cmd" : "move|setColor|...",
        "args" : "cmd arguments accordingly to driver description"
      }  
    ]  
  }  
}
```
Our Approach

- Discovery and connection

Diagram:

1. Discovery()
   1.1: LookupDriver()
   1.2: Connect()
   1.2.1: Ack()
   1.2.1.1: ConnectedToActuator()

2. Discovery()
   2.1: LookupDriver()
   2.2: DriverRequest()
   2.2.1: DriverResponse()
   2.2.1.1: Connect()
   2.2.1.1.1: Ack()
   2.2.1.1.1.1: ConnectedToActuator()
Our Approach

- Actuation sequence
M-Act

EventBus

S2PA  LocationService  ConectionService  AdaptationService  MEPAService  MACTService
M-Act
Register connected M-Objs

{ mhub_1, mobj_1, state1, connection1 }
{ mhub_1, mobj_2, state2, connection2 }
{ mhub_2, mobj_3, state3, connection3 }
...
{ mhub_X, mobj_Y, stateZ, connectionY }

Drivers connected M-Objs

{driverA}
{driverB}
...
{driverX}

RegControl / Redirect Msg

{ mhub_id, mobj_id, CMD[] } / response

Publisher

Subscriber

Universal DDS Interface
Driver

- **JSON format**

```json
{
    "device_type": "bb8",
    "interface": "ble",
    "connection": {
        "service": <uuid_service>,
        "characteristic": <uuid_characteristic>,
        "cmd": <cmd_array>,
    },
    "commands": {
        "roll": {
            "service": <uuid_service>,
            "characteristic": <uuid_characteristic>,
            "cmd": "0x30",
            "arg_size": "5",
            "arg_type": "byte",
            "offset": <offset_expression>,
            "cmd_line": "0xFF, 0xFE, <cmd>, <args>, <arg_size>, <offset>"
        },
        <list_other_commands_available>,
    }
}
```
Driver

- JSON format

Device type and communication interface it uses

Connection Steps (e.g. To unlock the device)

List of available commands in a generic format and the specification to translate to the native protocol

```json
{
  "device_type" : "bb8",
  "interface": "ble",
  "connection": {
    "service": <uuid_service>,
    "characteristic": <uuid_characteristic>,
    "cmd" : <cmd_array>,
  },
  "commands": {
    "roll": {
      "service": <uuid_service>,
      "characteristic": <uuid_characteristic>,
      "cmd" : "0x30",
      "arg_size" : "5",
      "arg_type" : "byte",
      "offset" : <offset_expression>,
      "cmd_line" : "0xFF, 0xFE, <cmd>, <args>, <arg_size>, <offset>"
    },
    <list_other_commands_available>}

}
Tests

▪ Demonstrate the feasibility of our approach
▪ Control the Toy Robot BB-8
▪ Evaluate the overhead added
  • Average ~36ms added to translation
Conclusions and Future Work

- Demonstrated the feasibility of the approach
- Mobile gateways can incur in actuation failures due to disconnection
- Suitable for smart homes, smart buildings, and systems where realtime are not required
Conclusions and Future Work

- Handover

{ Obj: BB8, M-Hub: A, Descriptor: {...} }

{ Obj: BB8, M-Hub: B, Descriptor: {...} }

M-Hub A

M-Hub B

SDDL

Gateway

Gateway

M-Hub

M-Obj

Joystick

BB8
Thank You!