Overview

• Me:
  • PhD Student at Mobile & Cloud Lab, since 2016
• Edge Process Management introduction
• Existing Simulation tools and options
• STEP-ONE - overview and case study
Service-Oriented IoT

• The inherent IoT problem is the interoperability issue
• Service-Oriented Architecture (SOA) to solve interoperability
• SOA standards
  • Web of Things (HTTP, JSON)
  • Resource Description Framework (RDF), JSON-LD, Web Ontology Language (OWL) - describe devices and their relationships
  • Discovery (WoT, SPARQL)
Composing Service-Oriented IoT

• 2nd issue is resource planning and coordination: operations, data-flows
• IoT scenarios dictate application processes which can may several organizations, their devices, while providing a high degree of personalization to end user
• **Business Process Management (BPM)** - the practice of observing, designing, analysing, automating and improving business processes
• BPM has seen significant success in enterprise software
BPM for IoT

- BPMN 2.0 standard
  - Visual standard
  - Machine-readable, executable
- BPMS-integrated IoT research has covered:
  - modelling [9, 5, 1, 8]
  - adaption to unexpected events and context changes [7, 12, 10]
  - dynamic service discovery, composition [2, 4]
  - case studies [14, 13]: healthcare, as a notable example

**Figure:** Example BPMN 2.0 diagram
Towards Edge Processes

IoT has evolved from a centralised architecture with two layers: cloud and end-devices, towards de-centralized models with additional intermediate layers.

• Previously mentioned research targets the conventional architecture.

• Following the emergence of Edge/Fog Computing, there is a need to adapt BPMS to the edge devices: **Edge Process Management (EPM)**

• EPM brings benefits such as:
  • high performance in terms of communication latency
  • privacy, as the tasks and data involved in a process may be kept on local devices and network
  • resilience and fault-tolerance: in cases where infrastructural damage or outages occur, the edge process can continue enacting (consider disaster scenarios)
EPM Challenges

- Edge environment imposes a highly volatile context, consider the smart city environment:
  - network nodes may be mobile
  - intermittent connectivity, radio interference
  - constantly changing pool of resources and services

- While executing processes on end-devices such as phones has been researched [6, 3, 11], the evaluation scale is small.

- Large-scale experiments are costly

- Simulation is the common answer, but today, no existing simulation tool provides mobility-related aspects and process-awareness simultaneously!
Existing simulators

**Discrete-event simulation:** world state changes are caused by events occurring at discrete points in time. State between events is not captured.

- **SuMO**
  - Purely focused on mobility
  - Roads, traffic lanes, traffic lights, pedestrians, different vehicle types.
  - Can export generated mobility traces
  - Developed in C++

- **Ns-3**
  - Very powerful for networking
  - IP networks, wireless technology
  - Mobility with random way-point movement, no map-based movement
  - However, can import SUMO traces
  - C++, Python, considerable effort to configure and develop
Existing simulators (2)

- **OMNeT++**
  - Provides a core set of network modelling and simulation modules
  - Several frameworks extending core OMNeT++ exist such as INET, VEINs
  - INET has powerful models for wired and wireless technology, trace-based mobility models (SUMO)
  - VEINs is a framework for inter-vehicular communication based on SUMO
  - C++ development

- **ONE - Opportunistic Network Emulator**
  - Developed for opportunistic networks and routing protocols there
  - Map-based movement models, with user behaviour modelling (e.g. workday)
  - Network modelling mainly at link layer
  - some physical layer aspects (interference, distance-based rate limiting)
  - Java-based
Opportunistic Network Emulator

We are extending ONE with process-oriented features, as:

- Large portion of existing BPMS software suites are Java-based (Activiti, Camunda, Flowable, jBPM, Bizagi)
- Detailed physical layer simulation is not feasible at large scale
- Good mobility modeling features

ONE core ideas:

- **Hosts** - agents moving in the world with networking capability
- **Messages** - hosts exchange messages
  - Can be consumed by applications embedded on hosts
  - Can forwarded according to routing protocol
  - New messages can be created by applications
Configuring and running simulations

Configuration-file based running

- Define hosts, host groups, applications, routing protocols, etc.
- Define map files to use (.wkt-based)
- Define mobility models to use

After a simulation run finishes, you can generate reports regarding networking statistic, encounters, etc.
Extension goals

- Embed a full-featured process engine on each simulated node, compliant with BPMN 2.0 (Flowable)
- Interface simulation world events with processes
  - Connection events as signals
  - ONE messages translated to process engine messages and vice versa
  - Simulated tasks and basic hardware modelling
  - Wired networking
  - Some other useful things such as "Go to (coordinates) task"
  - Set of BPM reports
Example scenario - Video capture from public transport

- Capture state of streets and roads with video processing
  - Buses with dash-cameras
  - Fog servers deployed at bus stations
- Process:
  - if the central road monitoring system needs updated analysis of a road segment, it forwards this request to appropriate the bus station server.
  - when a bus approaches, the server forwards the request to the bus, starting as a new process instance
  - the process handles the video capturing, once the bus reaches the end of the road segment (at a 2nd bus stop), and delivers the video content to the 2nd fog server, ending the process
  - the fog server does video compression and then uploads it to the central system for analysis
Demo
Conclusion and direction

- Integrating the engine and BPMN features is mainly done
  - Messaging, signals, process variable support is there
  - Missing time-based events, location-based
  - Reports on processed started, completed, events and signals raised is there.
- Hardware modelling of processing tasks needs improvement
  - Basic single-core ”task processing” is there.
  - lack of multi-core / concurrency modelling
- Need to benchmark the scalability - so far have tried with up to 60 nodes
- Need to polish a convincing scenario
References

Modeling IoT-aware Business Processes.

A Middleware for Discovering Proximity-Based Service-Oriented Industrial Internet of Things.

A resource oriented integration architecture for the Internet of Things: A business process perspective.

Adaptable Service Composition for Very-large-scale Internet of Things Systems.

Using BPMN to model Internet of Things behavior within business process.


Supporting adaptiveness of cyber-physical processes through action-based formalisms.

[8] **Martins, F., and Domingos, D.**
Modelling IoT behaviour within BPMN Business Processes.
*Procedia computer science 121* (2017), 1014–1022.

The Things of the Internet of Things in BPMN.
References


A Lightweight Process Engine for Enabling Advanced Mobile Applications.

[12] Seiger, R., Herrmann, S., and Amann, U.
Self-Healing for Distributed Workflows in the Internet of Things.

Modeling, Enacting, and Integrating Custom Crowdsourcing Processes.

A Workflow-Based Mobile Guidance Framework for Managing Personal Activities.