On Research Challenges in IoT Systems Engineering

16 August 2018, USC

Schahram Dustdar

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at
Ecosystems of People, Systems, and Things
What exactly is the "Internet of Things"?

Smart Systems and the Internet of Things are driven by a combination of:

1. SENSORS & ACTUATORS
2. CONNECTIVITY
3. PEOPLE & PROCESSES
We are giving our world a digital nervous system. Location data using GPS sensors. Eyes and ears using cameras and microphones, along with sensory organs that can measure everything from temperature to pressure changes.
2 CONNECTIVITY

These inputs are digitized and placed onto networks.
These networked inputs can then be combined into bi-directional systems that integrate data, people, processes and systems for better decision making.
Network Neutrality (NN)

All traffic on the Internet must be treated equally.

Can new innovative solutions compete?
Real cases

T-Mobile Germany Blocks iPhone Skype Over 3G and WiFi

James Kendrick Apr 6, 2009 - 5:15 PM CDT

Skype has been one of the top downloaded apps for the iPhone since its release last week, even though Apple (s aapl) bowed to AT&T (s T) in the U.S. to prevent the VoIP program from working on 3G. Users are reporting that they can’t use Skype on the iPhone, somewhat doubt concerning the software.

Netflix throttles its videos on AT&T, Verizon Networks

Streaming service says it limits video quality to protect users from exceeding data caps

By RYAN KNUTSON and SHALINI RAMACHANDRAN
Updated March 24, 2016 10:55 p.m. ET

AT&T Inc. and Verizon Communications Inc. were on the defensive last week after accusations swirled they were throttling the quality of Netflix Inc. video on their wireless networks.

It turns out it was Netflix that was doing the throttling.
Understanding the context
What about the Internet of Things (IoT)?

Prioritize partner device vendors

Prioritize partner IoT platforms
Diverse users with complex networked dependencies and intrinsic adaptive behavior – has:

1. **Robustness & Resilience mechanisms**: achieving stability in the presence of disruption

2. **Measures of health**: diversity, population trends, other key indicators
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem
Water Ecosystem
Air Ecosystem
Motivating Case Studies

**Building Management System**
- Manages building facilities, e.g., HVAC systems, elevators and emergency alarms

**Fleet Management System**
- Manages fleets of electric vehicles worldwide (e.g., on golf courses)
Motivation

• Lack of systematic support and tools for developing and operating IoT Cloud systems

• Today IoT Cloud systems are vertically closed and tightly coupled
  • Hard to develop and maintain applications
  • Difficult to operate and reuse existing infrastructure
Programming Model for IoT Cloud Applications
Motivation

Fleet energy usage management

Process energy consumption

Actuation steps to stop vehicle

- Detect energy fault
- Notify manager
- Stop vehicle

Requirements:

• **Application**: Custom configuration and behavior of Sens./Act.
• **Runtime**: Dealing with constrained resources
• **Developer**: Domain expert knowledge

Requirements:

• **Application**: Should be generic (independent of underlying devices)
• **Runtime**: Dealing with scalability and elasticity concerns
• **Developer**: Software engineering expertise

- Fleet energy usage management
- Process energy consumption
- Actuation steps to stop vehicle
- - Detect energy fault
- - Notify manager
- - Stop vehicle

- IoT Cloud Application
- Communication Channels
- Actuation Control Sequence
- Monitoring Data Processing
- Gateways
- Cloud Platform
- User
- Storage
- Core Application Business Logic
- Communication Protocols
  - CoAP
  - MQTT

- Low-level Communication Channels
- Actuators
- Sensors

Actuation steps to stop vehicle
Approach

- Task - Encapsulates domain-dependent controls or analytics
  - Packaged into domain-specific libraries (e.g., vehicles management)

- Intent - High-level representation of Tasks on Cloud platforms
  - Used by developers to remotely invoke Tasks
  - Independent of concrete Task implementation

Intent-based Programming Model

- Passive data structure which declaratively describes intended action, e.g., stop vehicle
- Generic applications (What needs to be done instead how to do it)
- Enable developing loosely coupled applications

- Trade expressiveness for more flexible and easier application development
Provisioning solutions for Smart Cities
“abstractions, concepts and processes“
Research Challenges Overview

**Research Challenges** addressed by **Design Principles** implemented by **Main Enablers**

- **On-demand, self-service usage models**
  - Enable dynamic feature composition
  - Support API encapsulation of the infrastructure resources

- **Unified representation of heterogeneous resources**
  - Enable managed configuration models
  - Enable resource monitoring

- **Cost-awareness**
  - Enable fine-grained resource consumption
  - Enable automation of provisioning processes

- **Efficient provisioning models**
  - Support elastically scalable provisioning processes
  - Provide more autonomy to the edge resources

- **Flexible customization of tightly coupled resources**
  - Logically centralized point of operation

- **Support for elasticity concerns**
  - Software-defined gateways
  - Multi-level provisioning workflows

- **Device profiler**
  - Centralized infrastructure API management
  - Cloud-based controller

- **Software-defined IoT topology (complex units)**
  - Edge-compatible provisioning agents
  - Flexible deployment and provisioning models

- **Configurations container**
  - Cloud-based dependency resolution
Motivation

Consider provisioning a simple application for monitoring environmental conditions in Smart City buildings

Current provisioning solutions:

- Require on site presence
- Require manual interaction with devices
- Not suitable for resource constrained devices

How to efficiently provision IoT Cloud applications?
Approach: Core principles

• From physically isolated, rigid Edge/IoT infrastructure to virtualized, elastic IoT Cloud, by utilizing “software-defined“ principles.

• From task-specific solutions to fully-fledged ecosystem and management processes, based on DevOps best practices.
Approach: Design Principles and Main Enablers

- Uniform representation of IoT infrastructure
- API encapsulation of infrastructure resources
- Logically centralized operation
- Automation of provisioning processes

Software-Defined Gateways

Provisioning Middleware

SDG

SDG

SDG
Approach
Software-Defined Gateways – Overview

• Execute atop physical gateways
• Virtualize gateways compute and memory resources

• Act as isolated containers for applications => lightweight execution environment

• Enable on-demand provisioning of application, libraries and configuration models

Software-Defined Gateways – Provisioning Model

• Independent of the underlying virtualization technology

• Built from SDG prototypes
  • Based on kernel-supported virtualization: LXCsv, libvirt-sandbox, chroot, etc.
  • Preconfigured with different functionalities, e.g., monitoring mechanisms

• Expose provisioning APIs used to deliver complex functionality

• SDG IoT Units enable encapsulating application components, libraries and configuration models
SDGs Ecosystem

- Hierarchical structure of SDG components and capabilities.
- Enables distributing SDGs & SDG IoT Units in a market-like fashion, e.g., via SDG AppStore.

Hierarchy of basic SDG components (partial view)
Software-Defined Gateways – Architectural View

• Provisioning Agent
  • Handling remote provisioning requests
  • Dynamically downloading application components
  • Local installation of application components
Software-defined Gateway - Example

Ready to be deployed on IoT devices such as physical gateways or cloudlets
C2: Approach

Provisioning Middleware

SDG  SDG  SDG
Provisioning Middleware

Cloud

- Multi-level Provisioning API
  - APIManager
  - SDGManager
  - Application Manager
  - MonitoringCoordinator
  - DeploymentHandler

Edge device/
IoT Gateway

- Provisioning Daemon
  - SDG
  - Monitoring Agent
  - Virtual Buffers Daemon
Example of SDG-driven Provisioning Process

Cloud
- SDG Instance
- Sedona VM
- Monitoring Application

API Call

Provisioning Controller

IoT
- Gateway
- Gateway
- Gateway
- Gateway

SDG
SDG Delivery Models

1. SDG
   Gateway Device

2. Script
   Cloudlet
   Download

3. SDG
   Gateway Device

Well known Repository

Download

Gateway Device

CDN

Gateway Device
IoT & Data Science – Research Challenges

IoT-driven ecosystems
IoT/Data/Application Orchestration
Osmotic Computing

- In chemistry, “osmosis” represents the seamless diffusion of molecules from a higher to a lower concentration solution.

- Dynamic management of (micro)services across cloud and edge datacenters
  - deployment, networking, and security, ...
  - providing reliable IoT support with specified levels of QoS.

Osmotic movement of MELs in Clouds, Edge, Things
IoT Data Sources

1. **Representation**: Structure and represent the data to facilitate multiple modalities, exploiting the complementarity and redundancy of different data sources.

2. **Translation**: Interpret data from one modality to another, i.e., provide a translator that allows the modalities to interact with each other for enabling data exchange.

3. **Alignment**: Identify the relation among modalities. This requires identifying links between different types of data.

4. **Fusion**: Fuse information from different modalities (e.g., to predict).

5. **Co-learning**: Transfer knowledge among modalities. This explores the field of how the knowledge of a modality can help or enhance a computational model trained on a different modality.
IoT Computational Unit

1. **MicroServices** (MS), which implement specific functionalities and can be deployed and migrated across different virtualized and/or containerized infrastructures (e.g., Docker) available across Cloud, Edge, and Things layers.

2. **MicroData** (MD), encodes the contextual information about (a) the sensors, actuators, edge devices, and cloud resources it needs to collect data from or send data to, (b) the specific type of data (e.g., temperature, vibration, pollution, pH, humidity) it needs to process, and (c) other data manipulation operations such as where to store data, where to forward data, and where to store results.

3. **MicroComputing** (MC), executing specific types of computational tasks (machine learning, aggregation, statistical analysis, error checking, and format translation) based on a mix of historic and real-time MD data in heterogeneous formats. These MCs could be realized using a variety of data storage and analytics programming models (SQL, NoSQL, stream processing, batch processing, etc.).

4. **MicroActuator** (MA), implementing programming interfaces (e.g., for sending commands) with actuator devices for changing or controlling object states in the IoT environment.
IoT Programming Patterns needed

1. **Decomposing IoT data analysis activities into fine-grained activities** (e.g., statistics, clustering, classification, anomaly detection, accumulation, filtering), each of which may impose different planning and run-time orchestration requirements;

2. Identifying and integrating **real-time data from IoT devices and historical IoT data** distributed across Cloud and Edge resources;

3. Identifying **data and control flow dependencies** between data analysis activities focusing on coordination and data flow variables, as well as the handling of dynamic system updates and re-configuration;

4. Defining and tagging each **data analysis activity with runtime deployment constraints** (QoS, security and privacy).
Grand Challenges Manifesto

Thanks for your attention

Prof. Schahram Dustdar
Member of Academia Europaea
IBM Faculty award
ACM Distinguished Scientist
IEEE Fellow

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at