Middlewares for Smart Objects and Smart Environments: Overview and Comparison

Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, Claudio Savaglio

2014

PUC RIO

Roadmap

- Introduction
- Middleware Requirements
- An Overview
- Comparison and Discussion
- Conclusion and Challenges

Introduction

- Internet Of Things (IoT):
  - Everyday objects will be connected to the Internet, will be identified, and will, possibly, communicate with other devices
  - These objects are typically referred as "**Smart Objects (SOs)**"
    - Equipped with hardware components: radio for communication, CPU to process tasks, sensors/actuators
  - These SOs are able to collaborate and to support Ambient Intelligence: **Smart Environments (SEs)**

Introduction

- Research is focused on defining new frameworks/middlewares for the rapid prototyping of SOs and SEs. To resolve many issues:
  - Communication among SOs
  - The interface with sensors/actuators
  - The proactivity
  - The knowledge management
  - The distributed computation

Middleware Requirements

- Requirements for SE middleware:
  - Req1: Abstraction over heterogeneous input and output hardware devices (to use a kind of “plug-and-play” paradigm)
  - Req3: Abstraction over data stream (to formalize data streams)
  - Req5: Abstraction over the development process (to analyze, design and implement smart environments using high-level abstractions

Middleware Requirements

- Requirements for SO middleware:
  - Req3: Management of Smart Objects
    - Tons of distributed SOs could potentially interact with each other and/or be used to fulfill a final goal
    - Discovery Services: dynamic properties
  - Req4: Evolution of Smart Objects Systems
    - Applications and SOs should be simply and rapidly prototyped and upgraded through proper programming abstractions
    - The evolution can be driven by programming and/or by learning
      - Evolution by learning is usually based on smart self-evolving components able to self-drive their evolution on the basis of some learning model

An Overview of Middlewares

- Robot Operating System (ROS)
  - "An Intelligent environment is very similar to a static, non-removable robot."
  - An adaptation of a framework on robotic domain: Player
  - Nodes: processors that perform computation and communication
- Intelligent Room (iRoom)
  - Track people in a Room
  - Use of Agents (SodaBot)

An Overview of Middlewares

- Gaia
  - LUA (Past member: Renato Cerqueira)
  - Distributed middleware infrastructure capable of managing resources cointaned in physical spaces
  - The framework provides mobility and adaptation

An Overview of Middlewares

- UbiComp
  - Supports the interaction of heterogeneous smart objects and allows to combine them through na GUI application
  - Artifacts: UbiComp components augmented with sensors, actuators, processing and network units. Differents properties (sensor reading)

An Overview of Middlewares

- Smart Products
  - Autonomous object
  - To support natural interaction between objects and users and to make smart products collaborating in the environment
  - Workflows for each procedure
  - Component-based architecture
    - Interaction between user and smart products
    - Cooperation between different smart products
    - Components for sensing, processing
    - Components for storing knowledge

An Overview of Middlewares

- ACOSO (Agent-based Cooperating Smart Objects)
  - Funded by the Italian Government
  - To provide a simple programming model to realize cooperating smart objects
  - Event-driven architecture: SOs to fastly react to external stimulus
  - Proactivity based on inference rules and on local and remote Knowledge Bases.

An Overview of Middlewares

- ACOSO (Agent-based Cooperating Smart Objects)
  - The architecture is platform independent (JADE, Jadex, MAPS). But it relies on the JADE, that provides effective agent management and infrastructure of communication
  - Behavior, EventDispatcher, Communication Management Subsystem, Device Management Subsystem (coordinating the sensor/actuator), KB Management Subsystem (managing the knowledge base in the SO)
  - Study case: Smart Office and Smart Body (smart chair, smart projector)

Comparison

- Middleware Requirements-Based Comparison

Table 1 Comparison of the overviewed SE/SO middlewares based on the general SE middlewares requirements

	SO/SE middleware	SE Req1	SE Req2	SE Req3	SE Req4	SE Req5
ROS-player/stage	SE	Yes	Yes	Yes	Yes	Yes
iRoom	SE	Yes	Yes	No	Yes	Yes
Aura	SE	Yes	Yes	Yes	No	Yes
Context toolkit	SE	Yes	Yes	No	No	Yes
JCAF	SE	Yes	Yes	No	No	Yes
Gaia	SE	Yes	Yes	No	No	Yes
Ambient agoras	SE	N/A	N/A	N/A	No	N/A
Voyager-2WEAR	SO	Yes	Yes	Yes	No	Yes
Smart-Its	SO	Yes	Yes	No	No	Yes
UbiComp/GAS	SO	Yes	Yes	Yes	No	Yes
FedNet	SO	Yes	Yes	Yes	Yes	Yes
Smart products	SO	Yes	Yes	Yes	Yes	Yes
ACOSO	SO	Yes	Yes	Yes	Yes	Yes

Comparison

- Middleware Requirements-Based Comparison

Table 2 Comparison of the overviewed SE/SO middlewares based on the specific SO middlewares requirements

	SO/SE middleware	SO Req1	SO Req2	SO Req3	SO Req4
ROS-player/stage	SE	No	Yes	Yes	Yes
iRoom	SE	No	No	Yes	Yes
Aura	SE	No	Yes	No	Yes
Context toolkit	SE	No	No	No	Yes
JCAF	SE	No	No	No	Yes
Gaia	SE	Yes	No	No	Yes
Ambient agoras	SE	N/A	N/A	No	N/A
Voyager-2WEAR	SO	No	Yes	No	Yes
Smart-its	SO	Yes	No	No	Yes
UbiComp/GAS	SO	Yes	Yes	No	Yes
FedNet	SO	Yes	Yes	Yes	Yes
Smart products	SO	Yes	Yes	Yes	Yes
ACOSO	SO	Yes	Yes	Yes	Yes

Comparison

- Middleware Features-Oriented Comparison
  - The programming model used (agent-oriented, service-oriented, component-oriented)
  - System Architecture (client/server, distributed, hierarchical, peer-to-peer)
  - Communication
  - Proactivity (i.e. autonomous)
  - Cooperation
  - SO/Application coupling
  - Discovery
  - Knowledge management

Conclusions

- For IoT, ACOSO, FedNet, Ubicomp and Smart Products are fully suitable to be actually exploited
- The control of 50 billions of objects cannot be subject to a human-driven management. So, self-* is necessary

Conclusions

- "MultiAgent Systems prove to be the right means to infuse intelligence into smart objects and in the whole big scenario
- "Agent-oriented methodologies could be exploited as basis for formalizing such an effective development methodology for Sos"