

Demo Abstract: Interoperable Mobile Agents in Heterogeneous Wireless Sensor Networks

Teemu Leppänen[†], Jose Alvarez Lacasia[‡], Archana Ramalingam[†], Meirong Liu[†], Erkki Harjula[†], Pauli Närhi[†], Jani Ylioja[†], Jukka Riekkilä[†], Kaoru Sezaki[‡], Yoshito Tobe[§], Timo Ojala[†]

[†] Department of Computer Science and Engineering, University of Oulu, Finland

[‡] Institute of Industrial Science, University of Tokyo, Japan

[§] RealWorld Communication Lab, Aoyama Gakuin University, Japan

teemu.leppanen@ee.oulu.fi[†], jpalvarez@mcl.iis.u-tokyo.ac.jp[‡],

yoshito-tobe@rcl-aoyama.jp[§]

ABSTRACT

We demonstrate interoperable mobile agents for language- and platform-independent wireless sensor network programming, with low-power resource-constrained embedded devices as static nodes and Android-based smartphones as mobile nodes, over disparate networks: 6LoWPAN and Wi-Fi. Representational state transfer architectural principles are applied in agent composition, control and migration and exposing the system resources to the Web: the devices, agents, sensor data, tasks and data processing results. The adaptable agent composition includes the task code in any programming language, the agent migrates according to a resource list and the state, i.e. the intermediate task result, represents the agent as a system resource. Mobile agents are then utilized in two tasks: first to collect light sensor data cooperatively in location, saving the mobile node battery whenever possible, and secondly the magnetometer sensor data with the scanned Wi-Fi access points' signal strength is used to detect groups of mobile nodes moving in the same direction in real-time.

Categories and Subject Descriptors

C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; D.1.3 [Concurrent programming]: Distributed programming

Keywords

Wireless Sensor Networks, Mobile Agents, Embedded Web Services, Constrained Application Protocol

1. INTRODUCTION

In wireless sensor networks (WSN), with limited hardware and software capabilities, the energy consumption and

efficient resource utilization are the main concerns. Traditionally, WSN programming has been either application- or domain-specific [1]. To enable flexible on-demand application deployment for WSN, the mobile agent -paradigm has been suggested [1]. Benefits include: reduced communication costs, asynchronous task execution, robustness, flexibility in the system configuration, system integration and co-operation capabilities [1, 6]. Mobile agents act autonomously, enable high-level abstractions of the system resources, enable direct manipulation of the device and possess self-properties [6].

This work is the first real-world application of the mobile agents in [3]. Previous work in programming WSN includes over-the-air programming, macroprogramming languages and virtual machines, notably the Java-based approach with mobile agents in [1]. Whereas we present novel language- and platform-independent adaptive mobile agent composition, with uniform interface for operating the agents and accessing the system resources, enabling the agent-based interoperability of heterogeneous resources-constrained embedded devices over disparate wireless networks.

2. SYSTEM OVERVIEW

The system architecture, in Figure 1, is based on the IETF CoRE framework and Representational State Transfer architectural principles for system resource access and for mobile agent composition, control and migration. We utilize resource directory [5] to store the system resource descriptions for lookups, including the sensors and data in nodes and the agents. Proxy abstracts the WSN as a single entity and translates HTTP requests to corresponding CoAP messages and vice versa. Code repository within the proxy hosts the agent's task codes, then accessible as system resources. The static WSN nodes [4] communicate with CoAP atop 6LoWPAN protocol stack with IEEE 802.15.4 radios on 868 MHz. As the mobile nodes, we utilize Android smartphones with integrated sensors, Samsung Galaxy S III, communicating with HTTP over IEEE 802.11 on 2.4 Ghz band.

3. USING THE MOBILE AGENT

The mobile agent composition in Table 1, includes a unique name for the agent for the resource lookups and consists of three segments: code, resources and state [3]. The code segment includes the data processing task code or its reference

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the author/owner(s).

SenSys'13, Nov 11–15, 2013, Roma, Italy.
ACM 978-1-4503-2027-6/13/11 ...\$15.00.

Table 1: Mobile agent composition

Name	sensing_agent
Code	room_light = moving_average(light); location = thresholding(wifi); features = extract_features(magnetometer);
Resource	coap://S_IPv6_addr/light http://1_IPv4_addr/light,magnetometer,wifi http://2_IPv4_addr/light,magnetometer,wifi http://3_IPv4_addr/light,magnetometer,wifi http://4_IPv4_addr/light,magnetometer,wifi
State (and local variables)	room_light = ... location = ... features = ... light = ... magnetometer = ... wifi = ...

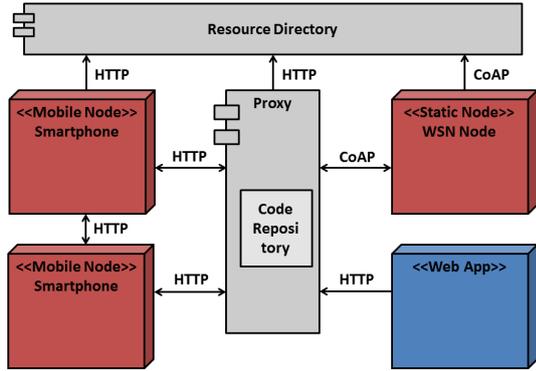


Figure 1: The system architecture.

with intended platform identifier, for on-demand lookup and retrieval from the code repository. The resource segment contains URLs representing the utilized resources, i.e. sensors and data types in the nodes. The state segment contains the current result of the task as the agent’s state, as well as the retrieved resource values as local variables.

The agent operates with the GET and POST methods. The agent migrates, as JSON data structure over HTTP and as CoAP message in the WSN [3], according to the resource segment URLs. When received by a hosting device, remote resource values are retrieved, the task is run and the agent state updated. Clients from the proxy and system devices directly can access the agent’s representation by its URL, i.e. name, from the hosting device. Thus the agent essentially provides a service in the system [3].

4. DEMO SYSTEM FEATURES

Figure 2 shows the demo setup. The proxy composes the agents from the dynamically available light and magnetometer sensors and Wi-Fi scanners. The agent is injected into the system as depicted with dotted line. Here the task code is precompiled IntelHEX binary for the static node and Python for the mobile nodes [3, 4]. The agent migrates as depicted by the solid lines from node to node based on the resource segment. The proxy is utilized to migrate the agent between networks. The agents operate followingly:

(1) Agents collect light sensor data and calculate the mov-

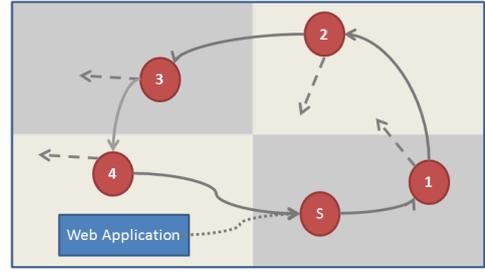


Figure 2: The application demo setup with a static node (S) and the mobile nodes (1-4).

ing averages of the data to provide coarse real-time lighting information in the location (colored background).

(2) Agents in mobile nodes scan Wi-Fi access points for their signal strength to determine the coarse location of the mobile node, by thresholding the signal strength with pre-determined thresholds. If a mobile node is detected within the coverage area of the static node (nodes S and 1 in Figure 2), the mobile node is requested to turn off the light sensor for saving the battery while in the area.

(3) Agents in mobile nodes collect magnetometer sensor data and filter it as in [2].

(4) The application then uses agents’ states to detect groups of mobile nodes moving in the same direction, by cosine similarity measure of the Wi-Fi scan measurements and magnetometer sensor data [2]. As an example, in Figure 2, the mobile nodes 3 and 4 move in the same direction as depicted with dashed lines.

5. REFERENCES

- [1] F. Aiello, G. Fortino, R. Gravina, and A. Guerrieri. A java-based agent platform for programming wireless sensor networks. *Comput. J.*, 54(3):439–454, 2011.
- [2] J. Alvarez Lacasia, T. Leppänen, M. Iwai, H. Kobayashi, and K. Sezaki. A method for grouping smartphone users based on wi-fi signal strength. In *Forum on Information Technology*, Tottori, Japan, September 4-6, 2013. [To appear].
- [3] T. Leppänen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Närhi, J. Riekkki, and T. Ojala. Mobile agents for integration of internet of things and wireless sensor networks. In *IEEE SMC2013*, Manchester, UK, October 13-16. [To appear].
- [4] T. Leppänen, J. Ylioja, P. Närhi, T. Rätty, T. Ojala, and J. Riekkki. Holistic energy consumption monitoring in buildings with ip-based wireless sensor networks. In *BuildSys’12*, pages 195–196, Toronto, Canada, November 6, 2012.
- [5] M. Liu, T. Leppänen, E. Harjula, Z. Ou, A. Ramalingam, M. Ylianttila, and T. Ojala. Distributed resource directory architecture in machine-to-machine communications. In *IEEE WiMob 2013 Workshop on Internet of Things Communications and Technologies*, Lyon, France, October 7-10, 2013. [To appear].
- [6] I. Satoh. Mobile agents. In H. Nakashima, H. Aghajan, and J. Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 771–791. Springer US, 2010.