Service Oriented Architecture for Heterogeneous and Dynamic Sensor Networks

Jérémie Leguay, Mario Lopez-Ramos, Kathlyn Jean-Marie, Vania Conan
Thales Communications
160 Bd de Valmy - BP82
92704 Colombes Cedex, France
Email: {firstname.name}@fr.thalesgroup.com

***Abstract*—The purpose of this demo is to foster a multi-level service oriented architecture for sensor networks that fully supports network dynamicity, auto-configuration, service discovery, and interoperability with legacy infrastructures. We consider a surveillance scenario in which the detection of an intruder, conducted within the range of a network of sensors, automatically triggers tracking activities.**

I. Introduction

The increasing complexity and heterogeneity of nowadays information systems have called for Service Oriented Architecture (SOA). Major benefits of such approach are modularity, flexibility, loose-coupling and interoperability. These information systems are increasingly connected to various kinds of sensors and actuators networks [1] having characteristics in processing power, battery, communication capability and availability that span over very wide ranges.

The main contribution of this demo is then the proposition and implementation of a multi-level SOA-based architecture for heterogeneous sensor networks. This architecture has two main goals. The first one is to extend SOA capabilities to devices with low capacities in processing power, battery, communication capabilities and availability. The second is to ease the deployment of network entities at all levels by providing auto-configuration mechanisms both at the network and service levels.

II. Service oriented architecture

The service-oriented architecture we propose classifies nodes in three different classes depending mainly on their available resources and network connectivity:

- **Full capacity nodes**: These entities have high availability and do not have processing power or battery issues. They could be critical always-online servers or client applications.
- **Limited capacity nodes**: These devices can be limited in terms of storage, battery, processing power or communicating capabilities but can still perform complex tasks and host operating systems such as Windows CE or Linux.
- **Low capacity nodes**: Such devices have extremely low capacity. They have few kilobytes of RAM, are equipped of a MicroControler and often use low power wireless interfaces such as IEEE $802.15.4$.

On the full capacity nodes, information is exchanged using common Web Services stacks (e.g., Axis). On the limited capacity nodes, we propose to use DPWS [2], which is perfectly adapted to dynamic and constrained devices and compliant to Web Services standards. At the lowest level, since to our knowledge no SOA-compliant service stacks were available, we propose to use a SOA-based solution (see Sec. III). The interoperability between Web Services stacks and the sensor network is ensured by a specific gateway (see Sec. IV).

III. SOA for low capacity nodes

As targeted devices in the category of *low capacity* devices are not able to process XML messages, we propose a tiny SOA stack which consists in a simple protocol and software architecture reproducing the architectural concepts and information exchanges of regular SOA implementation. The main goal is to make sensors very limited in capacity able to host services, discover the services of the others, announce their services, invoke services and subscribe to events.

We implemented a tiny SOA stack under TinyOS on Crossbow MICAz sensors having a MTS310 sensor board attached to their serial port which offers a variety of sensing modalities such as light, pressure, acceleration, temperature and acoustic. This hardware combination offers also two *symbolic* actuators such as a sounder and a set of 3 leds. MICAz nodes have very limited capacity in memory and processing power as they embed an Atmel ATmega128L microcontroller with 4KB of RAM and have 128KB of programmable flash.

IV. Bridge

This section presents the software stack we propose to run on nodes that bridge between full-capacity nodes and low capacity nodes in our service oriented architecture. Its main functions are:

- *To enable a tiny SOA to DPWS service translation*. This is done through hosted service-specific proxies which can be automatically generated from TinyOS module headers. This fine-grained operations enable, for instance, to trigger an actuator in a specific node. This provides a feedback mechanism to the network which is able to react to events and thus enhance sensing capabilities.
- *To provide a high-level interface* which hides the complexity of the underlying network. In a context in which

hundreds or thousands of these nodes can be deployed, a one-to-one SOA message translation offers poor control possibilities. By providing meaningful interfaces, the network can be seen as a single *macro-sensor*. The chosen data dissemination mechanism is publish-subscribe.

All these modules have been conceived as *bundles* within the Open Services Gateway initiative (OSGi) framework in Java. This gateway has been implemented to run on Crossbow's Stargate sensor network gateways. This device is equipped with an Intel PXA255 Processor running at 400MHz and has 64MB of SDRAM. It embeds a PCMCIA Wi-Fi card, a Compact Flash memory card, an ethernet port and a 2 serial ports, one of which hosts a MICAz mote to provide Zigbee connectivity.

Despite the fact that the bridge can offer one-to-one service translation, we propose to use a more efficient and relevant way of communication based on a publish/subscribe mechanism: data dissemination is performed through a virtual *pipe* divided in different channels, each identified by a unique *topic*. When a new sensor is available, its event sources are announced as topics on the pipe; when consumers subscribe to an existing topic, nodes are requested to publish information on it as soon as it is available.

V. Proof-of-Concept

A. Demonstration setup

We considered a surveillance scenario in which sensors forming an ad hoc network have been deployed to detect intrusions via seismic vibrations. These sensors communicate using an Wi-Fi ad hoc network through the Crossbow Stargate gateway with an Axis 213 PTZ camera and a laptop running a Command and Control application.

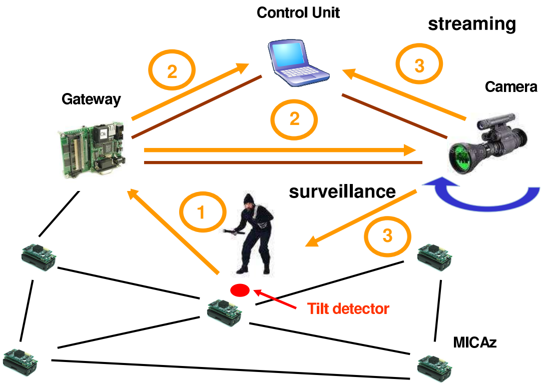
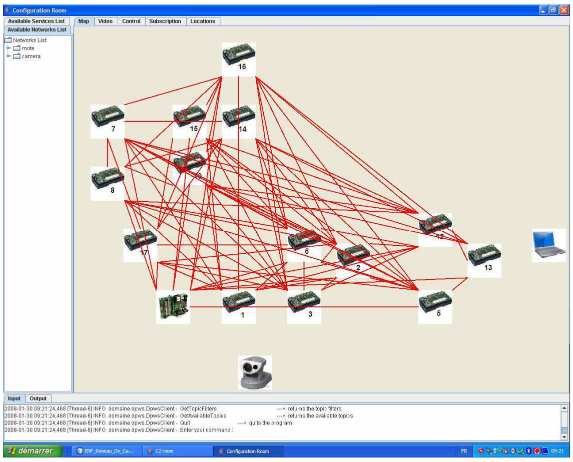


Fig. 1. Surveillance scenario.

Fig. 1 presents graphically the scenario that we considered and the sequence of events that take place when an intrusion is detected. Once routing has converged and nodes are able to exchange information, the SOA-based service stacks (both at the DPWS level and at the mote level) advertise their hosted services and topics which are then known to the rest of the network. The control unit then requests subscriptions to the topic providing tilt events from the accelerometer service of the motes. Whenever a mote detects a significant change in acceleration, a notification is sent through the bridge to the control unit and the camera, which react accordingly.

B. Command and Control unit

More specifically, the Command and Control unit that we have developed allows to manage all the sensors and actuators within the scenario. The application allows (1) to locate the different entities on a map and to see the network connectivity between them (see Fig. 2), (2) to plot graphs showing the evolution of measurements (acceleration, light, etc.) over time and to push an available event into the publish/subscribe pipe, (3) to watch the video streamed from the camera and set the preset positions corresponding to every monitored zone, and (4) to subscribe to topics available in the publish/subscribe pipe.



```
2008-01-30 09:21:24,468 [Thread-6] INFO  domaine.dpws.DpwsClient -  GetTopicFilters            ---> returns the topic filters
2008-01-30 09:21:24,468 [Thread-6] INFO  domaine.dpws.DpwsClient -  GetAvailableTopics         ---> returns the available topics
2008-01-30 09:21:24,468 [Thread-6] INFO  domaine.dpws.DpwsClient -  Quit          ---> quits the program
2008-01-30 09:21:24,468 [Thread-6] INFO  domaine.dpws.DpwsClient -  Enter your command :
```

Fig. 2. Command and Control unit GUI.

VI. Conclusion and Future Work

This paper has proposed a multi-level service oriented architecture for sensor networks. This architecture bridges the gap between devices having very different capacities and fully handle network dynamicity by providing auto-configuration features at both network and service level.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] "Devices Profile for Web Services (DPWS) specifications," http://schemas.xmlsoap.org/ws/2006/02/devprof/.