

# Toward the Next Generation of Sensors as a Service

Dario Lombardo, Vito Morreale and Giuseppe Li Calsi

**Abstract** This chapter presents a novel approach for service oriented architectures applied to the Sensors-as-a-Service paradigm. This work illustrates a flexible, scalable programming model of applications based on a service platform. The major contribution of this work is a new idea for service model, SNS model, compliant to sensor, where service is designed to use the information coming from sensor, and it is possible to their integration. The service model allows to optimize information that are coming from service or from sensor versus a new concept Service SNS.

## 1 Introduction

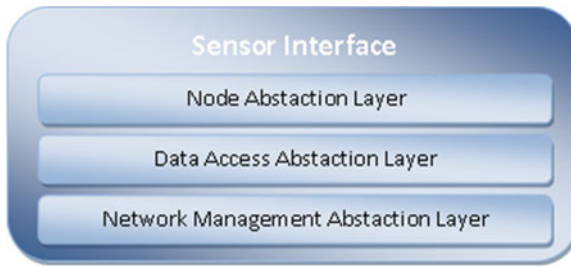
One of the most known and well-established development and deployment approaches to software systems is Service oriented architecture (SOA). It indicates a specific type of distributed system, in which the entities that constitute it are precisely the services. An emerging trend in recent years makes various types of Web-resident sensors, instruments, imaging devices, and repositories of sensor data, discoverable, accessible and controllable via the World Wide Web. It would be interesting to combine the two different aspects, i.e. service-orientation and sensor-as-a-service, into a unique services platform conceived, specialized, and optimized for sensors. This chapter aims at this goal, i.e. a new idea to develop a platform for ease of use and the integration of sensors as services.

---

D. Lombardo (✉) · V. Morreale  
Engineering Ingegneria Informatica spa, Viale Regione Siciliana, 7275, 90146 Palermo, Italy  
e-mail: dario.lombardo@eng.it

V. Morreale  
e-mail: vito.morreale@eng.it

G. Li Calsi  
demetriX srl, Piazza Stazione, 60, 90044 CARINI, PA  
e-mail: giuseppe.licalsi@demetrix.it



**Fig. 1** Sensor interface

This chapter is organized as follows: first a brief state of the art is presented (Sect. 2); then our solution is presented: Model SNS Service (Sect. 3), SNS platform (Sect. 4).

## 2 Service Oriented Architecture for Sensor Networks

One major reason for the increasing interest in wireless sensor networks (WSN) in the last few years is their potential usage in a wide range of application domains. This interest brings to the identification of the new paradigm of Sensors as a Service, to employ the same concept of service at sensor with the suitable correction for this case. The appropriate levels of abstraction [1–4] contribute to obtain the required specifications (Fig. 1):

- The *Node Abstraction Layer* handles information on sensory capabilities of sensor nodes providing mechanisms for uniform access by the higher levels.
- *Data Access Abstraction Layer* is related to collect data from networks sensory and move to higher levels according to the specifications
- *Network Management Abstraction Layer* provides an interface to the functionality most closely related to the management system of the network (setting, state nodes, etc.)

At the application level, each sensor may be modelled according to a service-oriented design [5–7], i.e. as including a set of actions that may be demanded to the sensor, with the help of services called by means of the framework, where services were developed. Those actions concern primarily querying the sensor in order to acquire the physical quantities under monitoring, and configuring the behaviour of the sensor by adding the new functionalities offered by specialized services.

The logical model of the sensor has to be agnostic with respect to its physical characteristics and to the measured quantity, thus allowing a wide range of flexibility whenever new sensors are added. There are some necessary features that a Service-Oriented Middleware for WSN would be required to implement. They include functional and non-functional requirements. In many application domains specific functions are essential since they derive from the application logic itself.

A middleware common to all applications has to include all major core services, in order to avoid that these should then be integrated in each application developed for the specific domain. The common requirements for such services are [8–10]:

1. Runtime support for the delivery of services and their execution
2. Support to consumers for the discovery of registered services
3. Transparency for client applications
4. Abstraction to hide the heterogeneity of environment sensors
5. Configurable services
6. Support to the mechanisms of self-organization
7. Interoperability with a wide range of devices
8. Efficient management of large volumes of data with high throughput communication
9. Cooperative processing of tasks should lead to more precise results and new application fields
10. Sensor networks require security mechanisms that are adaptive to environmental conditions
11. Support for the requirements of QoS
12. Support for integration with other software systems
13. Sensor nodes must perform tasks of network maintenance
14. All algorithms and protocols must be energy optimized.

Finally, service-oriented approaches shape the sensors as a series of services, allowing then to adopt the common and powerful paradigm of service-oriented architecture for building applications. The major strong point of the service-oriented middleware is the ease with which you can add advanced features simply by implementing new services.

### 3 SNS Service Model

The SNS model, see Fig. 2, is defined to represent SNS service with its structure and its attributes. The SNS model would like to convey a possible representation of sensor as a service. The model’s goal is to create an abstraction of the sensor, the physical layer, considering it as a service, and a software level, for interacting with sensor like if it was a service, by adopting the whole service-oriented paradigms. In the SNS Service model a connection between service and sensor is expressed formally, where Service SNS finds an integration at high level. It is possible to represent this formalization:

$$\begin{aligned}
 \textit{Service SNS} &= \textit{Sensor}(\textit{data}, \textit{dependence}) \\
 &+ \textit{Service}(\textit{data}, \textit{operation}, \textit{dependence})
 \end{aligned}$$

The SNS service model is an optimized conjunction of service and sensor, where the information related to both are represented. The SNS service model is composed

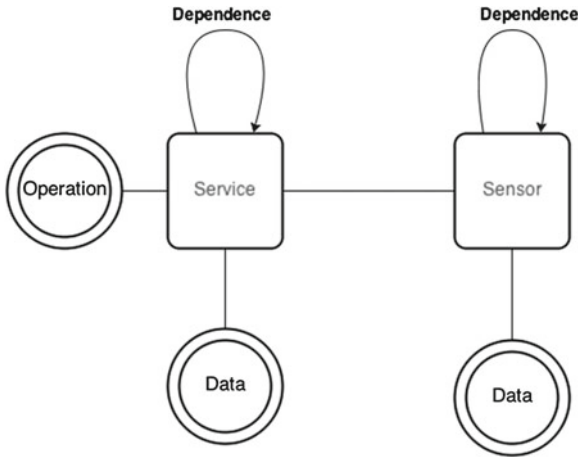


Fig. 2 SNS service model



Fig. 3 SNS service

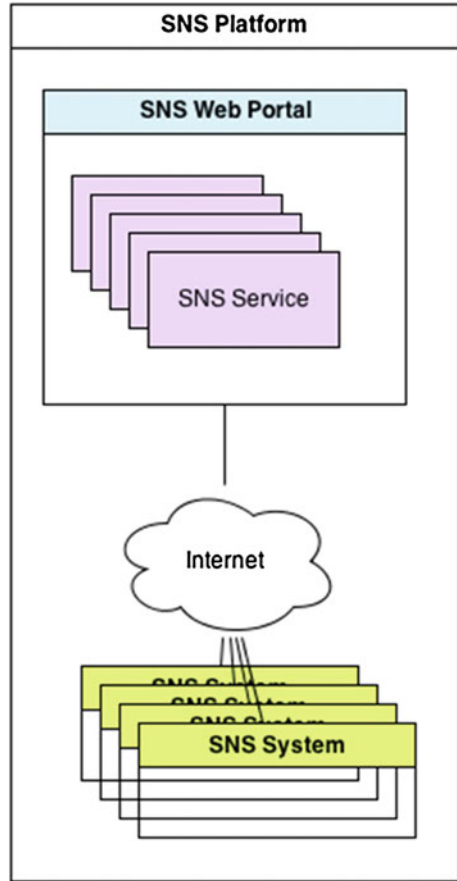
of a service-related part and a part more specific to the wrapped sensor. The service part includes information about operations and data related to them, whereas the sensor part specifies only the data to gather from the sensor; however a dependence relationship between service and sensor can be specified.

The *operation* attribute specifies what the service can do and its interaction with the external world, while the *data* attribute represents what information the service/sensor manage. The above-mentioned *dependence* concerns with the connection between services and sensors and allows to wrap a combination of several services or sensors with a SNS service.

### 4 Service Platform

This work proposes a standardized configuration of SNS service that strictly depends on its description. SNS service (Fig. 3) will exhibit an unique Web service interface through which client services and systems can exploit sensor's services.

Fig. 4 SNS platform



The SNS service has available all data necessary for its operation: such data come from the data supplied by sensors on network or other SNS services. So that SNS service can be properly configured by the user or application supposed to use it.

A SNS service (Fig. 3) is a software component that can be plugged into the platform to offer only one interface, to manage indifferently services and sensors. The SNS service, Fig. 4, will access to the sensors managed by the SNS platform through the sensor interface, being able to achieve measurements or perform any operation specified in the interface.

The information that the SNS service can exchange varies from service to service. Each SNS service makes accessible its description, by which it explains the dependencies (data sources), the operations and the data to be known, so that it can offer its services. The configuration of each SNS services will expose a standard WS management interface, that it is fixed by platform. This choice allows to have a single interface shared by all SNS services, providing the description of the SNS service, to carry out the basic operations and the configuration and the operations offered

by the SNS service. Nevertheless, the service will expose WS customized interface with its functionalities and operations, to make available outside of the platform. It is possible to access these services only after having carried out the configuration from the platform or through the management interface of the service, and obtaining its identification. The choice of imposing a single interface for managing all the services of the platform will result in releasing a single library for managing all services and that can be used by all modules of the platform.

In order to make the development of new SNS services as fasted as possible, a framework, called SNSWSFramework, has been defined and released, which contains a set of feature.

## 4.1 Service SNS

Each SNS service has its own XML description that contains the list of dependencies, the list of data to configure and list of offered operations.

**Dependencies:** Dependencies can be of two types: “sensors” and other “SNS services”. For each of them a name is given, along with the type of measure and the unit of measure. Some dependencies can be selected among several alternatives. This type of dependencies are described with the element `< dependencyChoices / >`.

**data:** For each data set it is possible to indicate the name, the unit of measurement, whether it is required to configure and, in case it is not present, its default value.

**Operations:** For each operation offered by the service, this field provides the name, the list of input data provided by the operation and the list of output data. For each of the data input / output the same information provided for the data service configurations is shown.

For use the SNS Service is necessary configure. The SNS Service configuration, in relationship of the own description, permit to build a new instance of specific SNS Service, and it is stored into definite repository.

**Configure SNS services:** Each SNS service needs to know in advance the information to access the data required for its operation. This information varies for each user and should be saved in an appropriate user configuration. When the user wants to use the services offered by SNS service, he or she has to provide the ID of this associated configuration. The information that can be configured for each service are of two types: dependencies and data.

The configuration management can be done through the management interface of the service using one of the methods exposed for configuration management:

- `editServiceConfiguration`: to edit a configuration already existing on the system, replacing the old one with the new one, by providing the identifier of the old configuration and the element `<serviceConfiguration />` with the new one; it returns the identifier of the new configuration in case of success;
- `getServiceConfiguration`: to retrieve an existing configuration, by taking the identifier of the configuration as input and getting back an element of type `<serviceConfiguration />` with the corresponding configuration.

**Configuring dependencies:** SNS Services offer functionalities to get data from sensors of the user or other SNS services configured by the user. In order to access this data, the service needs to know the required parameters that it is possible to distinguish according to the type of dependencies. The dependencies that need to be configured for the service SNS are indicated in its description; for each of them the type of measurement returns is indicated, and the unit of measure. To complete the configuration of dependency, it is necessary to specify a data source that returns a data consistent with the description. As previously mentioned, the dependencies are of two types, i.e. sensors and other SNS services, whereas configuration parameters of the dependence are:

1. *sensors dependencies:* to specify the data needed to access the instance of the SNS system that manages the sensors (address on which the service responds) and the identifier of the sensor;
2. *dependencies on other SNS services:* to specify the data needed to access the management interface of the service, the name of the operation of interest, the *id* of the configuration to be used;

**Configuration data:** The SNS services may require some parameters that have to be configured for their correct operation. These are also indicated in the description of the service and for each of them the necessary information to their configuration is given.

## 5 Conclusion

This chapter describes a new approach for developed services connected with a sensor. It defines a structure of an abstract service and the list of requirements for this service model. The information generated by a service and a sensor is handled by SNS platform: the source information is different every time for every sensor, but the interface to interact with the platform is uniform.

This work allows to be compliant with a service approach to distributed systems (such as sensor networks). The resulting API and platform are released according to the open source rules.

**Acknowledgments** This work is partially supported by the Italian Ministry of Economic Development through the project SeNSori (“SeNSori– Sensor Node as a Service for Home and Buildings Energy Saving”).

## References

1. Girolami, M., Lenzi, S., Furfari, F., Chessa, S.: SAIL: A Sensor Abstraction and Integration Layer for Context Awareness. University of Pisa, Pisa (2008)
2. Escolar, S., Isaila, F., Calderón, A., Sánchez, L.M., Singh, D.E.: SENFIS: a Sensor Node File System for increasing the scalability and reliability of Wireless Sensor Networks applications. J. SuperComputing (2009)

3. Escolar, S., Carretero, J., Isaila, F., García, F.: Deconstructing the wireless sensor networks architecture. In: IEEE Symposium of Embedded Systems [IES' 2006], Antibes, Juan- Les-Pins, Francia (October 2006)
4. Suba, F., Prehofer, C., van Gurp, J.: Towards a Common Sensor Network API: Practical Experiences. Nokia Research Center, Helsinki (2008)
5. Golatowski, F., Blumenthal, J., Handy, M., Haase, M., Burchardt, H., Timmermann, D.: Service-oriented software architecture for sensor networks. Institute of Applied Microelectronics and Computer Science, University of Rostock, Rostock (2003)
6. <http://www.opengeospatial.org/ogc/markets-technologies/swe>
7. Amundson, I., Kushwaha, M., Koutsoukos, X., Neema, S., Sztipanovits, J.: Oasis: a service-oriented middleware for pervasive ambient-aware sensor networks. Technical report, Institute for Software Integrated Systems Department of Electrical Engineering and Computer Science Vanderbilt University, Nashville, Tennessee, USA (2006)
8. Anastasi, G.F., Bini, E., Romano, A., Lipari, G.: A service-oriented architecture for qos configuration and management of wireless sensor networks. In: 15th IEEE International on Emerging Technology and Factory Automation, Bilbao (2010)
9. Anke, J., Muller, J., Spieß, P., Chaves, L.W.F.: A service-oriented middleware for integration and management of heterogeneous smart items environments. In: Processing of 4th MiNEMA Workshop, Lisbon (2006)
10. Caporuscio, M., Raverdy, P-G., Moun gla, H., Issarny, V.: ubisoap: a service oriented middleware for seamless networking. In: 6th International Conference on Service Oriented Computing: ICSOC, Stockholm (2008)