

Data-Centric UML Profile for Wireless Sensors: Application to Smart Farming

Julian Eduardo Plazas, Universidad del Cauca, GIT, Popayán, Colombia

Sandro Bimonte, Irstea, UR TSCF, Aubière, France

Gil De Sousa, Irstea, UR TSCF, Aubière, France

Juan Carlos Corrales, Universidad del Cauca, GIT, Popayán, Colombia

ABSTRACT

Modelling WSN data behaviour is relevant since it would allow to evaluate the capacity of an application for supplying the user needs, moreover, it could enable a transparent integration with different data-centric information systems. Therefore, this article proposes a data-centric UML profile for the design of wireless sensor nodes from the user point-of-view capable of representing the gathered and delivered data of the node. This profile considers different characteristics and configurations of frequency, aggregation, persistence and quality at the level of the wireless sensor nodes. Furthermore, this article validates the UML profile through a computer-aided software engineering (CASE) tool implementation and one case study, centred on the data collected by a real WSN implementation for precision agriculture and smart farming.

KEYWORDS

Aggregation, Data-Centric, Model-Driven, Node-Level, Precision Agriculture, Smart Farming, UML Profile, Wireless Sensor Networks

INTRODUCTION

The Agri-food sector plays a key role in the economy of almost every country in the world, not only for generating wealth and creating employment but also for the nutrition of the population in developed and developing countries (Lehmann, Reiche, & Schiefer, 2012; Ramirez-Villegas, Salazar, Jarvis, & Navarro-Racines, 2012). Different aspects, like increasing the sector profitability, adapting to the climate change, supplying the demands for emerging markets, or ensuring the products quality are currently challenging the Agri-food sector. Therefore, innovations as smart farming, precision agriculture or product tracking are vital for overcoming these challenges (Akanksha Sharma, Barbara Arese Lucini, Jan Stryjak, & Sylwia Kechiche, 2015; Lehmann et al., 2012; Plazas & Corrales, 2017; Ramirez-Villegas et al., 2012).

Such innovations rely on the intensive monitoring of the products and their environments, since the collected data allow for the detection of undesired situations, and the development of accurate information and forecasting systems. These complex systems are usually underpinned on complex simulation models calculated in real-time, which must rely on high-quality sensors data. Indeed, the advent of low cost sensors enabled the development of small sensing platforms with

DOI: 10.4018/IJAEIS.2019040102

This article, originally published under IGI Global's copyright on April 1, 2019 will proceed with publication as an Open Access article starting on February 4, 2021 in the gold Open Access journal, International Journal of Agricultural and Environmental Information Systems (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

wireless connection capabilities (sensor nodes), which can be gathered and deployed as Wireless Sensor Networks (WSN) to monitor areas where wired connections are difficult or inadequate to establish (Wang, Zhang, & Wang, 2006). These WSN are one of the most important Information and Communication Technologies (ICT) for smart farming and numerous other applications domains since they provide right-time crucial data from the monitored environment (Lehmann et al., 2012; Plazas & Corrales, 2017; Plazas, López, & Corrales, 2017).

However, handling agricultural collected data is challenging since the monitoring sensors can collect and stream large amounts of raw data (e.g. embedded in tractors) and must deal with limited and depletable resources (e.g. deployed on the crop fields) (Anisi, Abdul-Salaam, & Abdullah, 2015; Jabeen & Nawaz, 2015). These big data heterogeneous streams must be correctly and timely processed in order to serve for the different applications aiming to improve the decision-making, control and definition of strategies in the Agri-food sector or any other domain, considering the end-user needs. Especially, WSN data processing and analysis is crucial in smart farming to handle complex agricultural applications, such as phenology monitoring, yield estimation or environmental risk assessment (Shao, Ren, & Campbell, 2018). Moreover, the deployment of such composite system using WSN, information systems, simulation models, etc., often leads to architectural complex ICT solutions, whose design, implementation and maintenance can be difficult and expensive.

Overcoming these issues is a challenging task. Therefore, an effective design of the WSN is the first mandatory step to grant a high-quality implementation of such complex systems according to decision-makers analysis needs. Hence, conceptual modelling has strong relevance and wide acceptance since it allows to build solutions for real complex tasks apart from the implementation problems and limitations (Abrial, 2010).

In this context, the Unified Modelling Language (UML) is one of the most powerful tools for formalizing conceptual models, a widespread extensible object-oriented standard that closes the gaps between designers, developers and final users (Bimonte, Schneider, & Boussaid, 2016). However, to the best of our knowledge, current approaches do not provide a complete and effective conceptual representation of Wireless Sensor node (WS) data, which makes difficult to design complex Agri-food applications and reduces the applications' capacity to completely supply the end-user needs (Marouane, Duvallet, Makni, Bouaziz, & Sadeg, 2017; Paulon, Fröhlich, Becker, & Basso, 2014; Prathiba, Sankar, & Sumalatha, 2016; Thang, Zapf, & Geihs, 2011; Uke & Thool, 2016).

Considering this scenario, in this work, we propose a data-centric UML profile for WS. Our profile enables the modelling of different WS implementations from the gathered/available data characteristics, allowing for the definition of ICT applications capable of answering the user requirements. Moreover, among the different sensors computation methods, in this paper, we focus on data aggregation since it is useful for complex applications and necessary for saving the battery life time of WSN. Though we have placed our profile in the Agri-food domain, considering smart-farming applications, it is general enough to model the data behaviour of any Internet Protocol Smart Object (IPSO) -compliant sensing platform ('Smart Object Interoperability,' n.d.).

The remainder of this article is arranged as follows: the next section presents the main characteristics, configurations and types of data to consider for a WSN abstraction. Section 3 presents the state of the art, describing different types of aggregation in WSN and highlighting relevant works that could be leveraged alongside our profile in order to design and configure the most important layers of a WS-based application. Section 4 presents our data-centric meta-model, including the UML profile with some theoretical examples and its implementation in a CASE tool. Section 5 presents the profile validation within a real smart-farming WSN application. Finally, section 6 presents our conclusions and proposed future works.

WIRELESS SENSOR NETWORKS

In this section, we introduce the concepts of sensors, sensing probes, sensor nodes, and sensor networks. Furthermore, we state and describe some of the most important data characteristics and configurations for the definition of sensing applications.

A sensor can be any device capable of representing physical world conditions as measured data (Aqeel-ur-Rehman, Abbasi, Islam, & Shaikh, 2014). The basis for sensors are special materials that change their physical properties (*e.g.* their electrical resistance) with the environmental conditions (*e.g.* light, temperature). The most basic sensors (probes) simply leverage the properties of these materials to deliver an analogue signal (*e.g.* a voltage or resistance change) as a measurement that can be analysed to estimate the physical condition. Although, more advanced probes can deliver digital data of two or even more measured conditions.

These probes usually require software and hardware platforms to transform the raw signals into readable data. Current advances on low-cost hardware and easily-programmable microcontrollers have allowed for the development of small platforms capable of gathering data from various probes and delivering it through different communication protocols. Thereby, a sensor network consists of a set of interconnected sensor platforms (nodes) which can measure their environmental conditions. At the beginning, these sensor networks relied on wired technologies. Later, with the advent of wireless technologies, WSN started to be more and more used to monitor areas where wired connections are difficult, expensive or inadequate to establish (Wang et al., 2006).

Different WSN application domains, like smart farming or precision agriculture, require to deploy the nodes (WS) in non-accessible areas, placed in open and uncontrolled environments, and relying on batteries as their only source of power. Therefore, WS should consider the use of energy-efficient techniques like entering into Sleep Mode or reducing the transmitted data to avoid battery waste. Indeed, due to their deployment areas and/or their number, changing batteries of WS is not feasible. Thus, evolved energy saving methods must be considered based on the regulation of data gathering and delivering to make a balance between operational lifetime and data value. This should also be associated to some quality-checking techniques for the data reliability (Aqeel-ur-Rehman et al., 2014).

Moreover, other limited resources in WS are the memory and processing. Nowadays, the storage memory limitation can be solved by the use of microSD cards. However, the use of these kinds of memory has an energy cost. Moreover, resources associated to the microcontroller such as programming memory (*i.e.* RAM -Random Access Memory- and Flash) and processing (processor frequency) are still limited. These latter limitations are related to the need to reduce WS economic cost in order to enable the deployment of a large number of them. This is an economic philosophy adopted since the definition of the concept of WSN at the beginning of the 2000's with, in an extreme case, a WS at the price of one USD (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). This is reinforced by the integration of WSN in the higher concept of the Internet of Things (IoT) where billions of electronic devices would be deployed and connected (Atzori, Iera, & Morabito, 2010). A synthetic definition of the IoT concept is as follows (Guillemin & Friess, 2009): "The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service." In WSN, another limitation to consider is the communication range of the WS which has an impact in the deployment strategy and cartography.

The data collection and management considerations are very important for the definition of WSN applications, since they allow to assess the future effectiveness and efficiency of the network. Thus, in order to model the applications from the user point-of-view, we have selected some relevant characteristics and configurations for the WS data.

Relevant Data Characteristics

- The measurement type
- The measured value (considering the unit)
- The measurement location
- The measurement time
- The measurement estimated quality
- The remaining battery (considering the unit)
- The measurement probe information (position and model)
- The link quality
- Separation between internal (unavailable) and external (available) data

These characteristics allow to describe the WS data beyond the sensed (measured) value. For instance, the spatio-temporal information enables a more accurate decision-making; energy and hardware information enables a dependability assessment; and the separation between node-internal (gathered) data and node-external (delivered) data allows to define operations (e.g. aggregation) over the gathered data that only modify the delivered data.

Relevant Data Configurations

- The frequency for delivering the measurements
- The duration of the delivering window
- The granule of the delivering frequency and window
- The amount of measurements delivered in a window
- The frequency for gathering the measurements
- The duration of the gathering window.
- The granule of the gathering frequency and window
- The amount of measurements gathered in a window
- The lifetime of each measurement.
- The granule of the measurements lifetime

These configurations make an important separation between the gathered data and the delivered data. Since the gathering, processing and delivering of the data have different energy costs, these operations should remain separated in the WS configuration. Moreover, since most agriculture-oriented WS implement energy-efficient strategies like the Sleep Mode, the WS could define different working cycles to gather, process or deliver the data. Then, the Frequency indicates how often the node executes the operation. The Window duration indicates the working cycle of an operation. The Amount indicates how many times is the operation executed in one working cycle. Finally, the Granule is a unit of time that modifies the Frequency and Window. For example, for a Delivering operation with a Frequency of 10, a Window duration of 60, an Amount of 20, and a Granule of “minute”, the node will deliver the data at a rate of 10 times per minute for the first two minutes of the hour, and then stop delivering data for 58 minutes until a new 60-minutes Window starts.

Furthermore, data aggregation is a very important technique in WSN since it allows to reduce the transmitted data, the central storage space and the sensor noise (Anisi et al., 2015; Aqeel-ur-Rehman et al., 2014; Jesus, Casimiro, & Oliveira, 2017). Therefore, we also consider some configurations for the execution of aggregation functions in the WS.

Relevant Data Aggregation Configurations

- The aggregation function

- The frequency for aggregating the measurements
- The granule of the aggregating frequency
- The length of the aggregating window
- The amount of measurements aggregates in a window

Since aggregation is a data processing operation, it can be configured in the same way than the gathering and delivering operations with the addition of the aggregation function (e.g. average, maximum, mode) configuration.

These data collection and management considerations help to model the data behaviour in agricultural WSN applications. However, they are not restricted to agriculture-oriented and smart-farming applications; thereby, these WS data features could be used to model WSN applications for various domains outside the Agri-food context.

RELATED WORKS

In this section, we identify the most relevant aggregation types for smart-farming applications from state of the art. Moreover, we search, classify, select and analyse the current research on WSN data modelling.

WSN Aggregation Types

Reducing the computational load in central servers, which process and analyse big data streams produced by (wireless) sensor networks in Agri-food-oriented smart-farming applications, could allow for faster and more accurate situation management. Considering the specific case of WSN (an interconnection of smart devices), they allow for a distributed processing, i.e. manage the WS limited but useful processing capabilities for analysing the gathered data in order to reduce the storage and computing overload in the central servers by delivering only highly-valuable data (Anisi et al., 2015; Bonomi, Milito, Zhu, & Addepalli, 2012; Jabeen & Nawaz, 2015). This initial analysis can be achieved through different kinds of data aggregation.

Data aggregation is the process of summarising the gathered data for its statistical analysis, obtaining a small highly-valuable set of data through simple operations. In the context of precision agriculture, early data aggregation is important for saving resources and analysing relevant events occurring in different spatial and temporal scales sooner than in the central servers (Pozzani & Zimányi, 2010). Therefore, through a systematic mapping study based on (Petersen, Feldt, Mujtaba, & Mattsson, 2008), we identify the different aggregation types in the context of WSN (where the aggregation is performed in the WSN architecture and its scope). Then, we have identified four kinds of aggregation scopes:

- Spatial aggregation: when the aggregation is performed in order to reduce the amount of data produced by sets of sensor nodes located in different geographical (spatial) positions.
- Temporal aggregation: when aggregation is provided by only one node of the WSN and is realized using data in temporal windows.
- Spatial and Temporal: when spatial and temporal aggregations are performed.
- OLAP (On-Line Analytical Processing) (i.e. statistical) aggregation: when sensor data can be aggregated considering different spatio-temporal and thematic granularities.

Furthermore, the aggregation is performed in five different network levels:

- Nodes, some aggregation is performed on each individual sensor allowing to provide sensor-level data. This is the most relevant case for our research since analysing the data inside the WS reduce the resource waste by completely distributing the processing load.
- Central Nodes, these nodes are acting as cluster heads due to their increased capabilities. This aggregation cannot provide individual-sensor data.
- Base Station, these are the gateways or sinks harvesting the WSN data and transferring it to the Internet. Aggregating data in this level does not allow to provide individual-sensor data.
- Central Server, this is the central repository of the data. Though it allows to aggregate sensor-level data, central processing quickly depletes the network resources.
- Network, where aggregation is used to reduce the signaling in the network. This case is irrelevant for our research since this level is not used for analysing the data.

Considering the results of our study, in this work we focus on the more elementary aggregation and its location implementation: nodes (WS) and temporal aggregation.

WSN Data Modelling

Moreover, the WSN data must meet the user and application requirements for a successful implementation. An accurate design in a direct-engineering process supported with conceptual metamodells (*e.g.* UML profiles) of the data processed by WSN could allow to seamlessly meet such requirements.

Thereby, we conducted a second systematic mapping study (Petersen et al., 2008) aiming to identify current advances on conceptual models for describing the data inside sensor nodes, considering the importance of temporal aggregation and UML representations (Table 1). Our study considered the following classification criteria:

1. Domain: the conceptual model is used to represent a specific application or it is a generic model for WS applications.
2. Meta-model: the conceptual model is described in terms of meta-model or not.
3. Design Level: the conceptual model describes the data or other issues regarding WSN.
4. UML profile: the conceptual model is represented as a UML profile with stereotypes, tagged values and OCL constraints.
5. CASE tool implementation: the conceptual model is implemented in a Computer-Aided Software Engineering (CASE) tool.
6. Sensors implementation: the conceptual model is implemented over existing sensor nodes.

The results for this study (Table 1) show that most researches relating sensors, data and UML are focusing on modelling applications using sensors or other kinds of models, rather than designing meta-models for describing the data in sensors or the sensors applications. Furthermore, less than the half of the identified models consider the problems and limitations of a specific domain; indeed, only one study focused on agriculture.

Such results (Table 1) evidence that formal standardized models for describing the sensors' data and applications are scarce. Moreover, most of the identified UML profiles are designed for specific domains, without including agricultural applications like smart farming. Thereby, considering our research context (user-oriented WS applications), the most relevant works for the definition of a UML profile for temporal aggregation of data in WSN nodes are the ones reporting a meta-model or UML profile for the modelling of WSN data or applications (Marouane et al., 2017, 2016; Prathiba et al., 2016; Thang et al., 2011; Thramboulidis & Christoulakis, 2016).

In the first place, Marouane et al. (2016) use UML to represent structural and behavioural information in sensor nodes for an Advanced Driver Assistance System (ADAS) application in order

Table 1. Classification of the identified research

Domain	Paper	Meta-model	Design level	UML profile	CASE tool Implementation	Sensors implementation
General	(Uke & Thool, 2016)	No	Sensors Simulation	No	Yes StarUML	No
	(Prathiba et al., 2016)	Yes	Data	No	No	No
	(Firlej & Kresse, 2016)	No	Application	No	Yes Enterprise Architect	No
	(Sicari, Grieco, Boggia, & Coen-Porisini, 2012)	No	Physical	No	Yes	No
	(Thang et al., 2011)	Yes	Application data	No	Yes	No
	(Meyer, Sperner, Magerkurth, & Pasquier, 2011)	No	Application	No	No	No
	(Hong, Lim, & Song, 2011)	No	Application service	No	No	No
	(Gu, Zhu, Xiong, & Ding, 2010)	No	Application	No	No	Yes
	(Mohan, 2009)	Yes	Physical	No	Yes Enterprise Architect	No
	(Idoudi, Duvallet, Sadeg, Bouaziz, & Gargouri, 2008)	Yes	Physical service	Yes	Yes	No
	(Smuda, Gerhart, Shing, & Auguston, 2006)	Yes	Network	No	Yes Generic Modeling Environment	No
	(Lee & Song, 2006)	No	Network	No	Yes Rhapsody	Yes
	(Y. Zhang, Chen, Wei, & Huang, 2006)	Yes	Physical network	No	Yes	No
	(Champeau, Dhaussy, Moitie, & Prigent, 2000)	Yes	Physical	No	Yes Argo	No
Driver Assistance Systems	(Marouane et al., 2017)	Yes	Physical application	Yes	Yes MagicDraw	No
	(Marouane, Makni, Bouaziz, Duvallet, & Sadeg, 2016)	No	Data	No	Yes MagicDraw	No
Industrial Systems	(Thramboulidis & Christoulakis, 2016)	Yes	Physical network application	Yes	Yes Papyrus	No
Early Warning Systems	(Cama-Pinto et al., 2016)	No	Service	No	No	Yes
Agriculture	(Luvisi, Panattoni, & Materazzi, 2016)	No	Application	No	Yes	Yes
Education	(Tanik & Arkun-Kocadere, 2014)	No	Service	No	No	No
Medicine	(Machado et al., 2012)	Yes	Service	No	Yes Enterprise Architect	Yes
Military	(Yu, Dong, & Feng, 2012)	No	Service Simulation	No	Yes StarUML	No
	(Rajanikanth, Narahari, Prasad, & Rao, 2003)	No	Simulation	No	No	No
Robotics	(Mae, Takahashi, Ohara, Takubo, & Arai, 2011)	No	Service physical	No	No	Yes
	(Wongwirat, Paelaong, & Homchoo, 2009)	No	Service	No	Yes	Yes
Visual Surveillance Systems	(Kenchannavar, Patkar, & Kulakarni, 2010)	No	Service	No	Yes Visual Paradigm	No
Satellite Navigation Systems	(Jeong, Park, Lee, Lee, & Kim, 2008)	No	Service	No	Yes	No

to reduce the system design complexity. The paper also proposes some design patterns for sensing, processing and control of sensor data, and taking actions in ADAS applications.

Secondly, Marouane et al. (2017) propose an evolution of their previous work (Marouane et al., 2016) with an extension of the standard UML profile for adding real-time definitions and constraints, proposing a more suitable profile for representing the structural and behavioural information of sensors in ADAS applications in a formal standardized language.

In the third place, Thramboulidis and Christoulakis (2016) provide a UML profile for OMA LWM2M (Lightweight machine-to-machine communication protocol) and IPSO standard IoT objects. The proposed profile constitutes an approach to automate the integration of mechatronic components in the IoT environment through the generation of the LWM2M layer, leveraging IoT protocols in the development process of manufacturing systems.

Later, Prathiba et al. (2016) gather existing approaches that address data quality in WSN, defining three different models:

- Dataflow-level, where the data comes from the data source through aggregation and fusion points to the data sink.
- Group-level, where the sensor nodes are grouped and modelled as a whole, considering communication and aggregation operators.
- Node-level, which defines different tasks (sampling, sending, fusion, aggregation, etc.) according to the role of the sensor node in the WSN topology.

Finally, Thang et al. (2011) propose a UML meta-model for developing WSN data-centric applications. This meta-model allows to sample data from the probes, receive and forward data from different nodes, and process the in-node data with different rules sets. The authors also define a rule-execution engine and mention a model-to-text transformation for the implementation in sensors.

These related works evidence that the advances on modelling sensor nodes are very important since they reduce the design and implementation complexity in different application domains like driver-assistance and automated cyber-physical systems. However, the existing models and meta-models do not allow for a complete and discrete description of the sensed (unavailable) data and the delivered (available) data. Furthermore, the design of in-node data processing considering aggregation and quality for WSN monitoring applications (*e.g.* smart farming) is not supported by existing works (Jesus et al., 2017).

Therefore, since we have made special focus on agricultural WSN, we can conclude that our Data-centric Wireless-Sensor UML profile will have strong relevance in the definition of new smart-farming applications aiming to improve the Agri-food sector processes. Although, it could also be relevant in different domains like environmental monitoring or early warning systems.

DATA-CENTRIC WIRELESS-SENSOR UML PROFILE

In this section, we present our UML profile illustrated with different agriculture-oriented use examples (subsection 4.1), and its implementation in the commercial CASE tool MagicDraw (subsection 4.2).

The purpose of UML profiles is to allow customizing UML for particular domains or platforms by extending its meta-classes (class, property, etc.) (OMG, 2011). A profile is defined using three key concepts: stereotypes, tagged values and constraints. A stereotype extends a UML meta-class and is represented using the notation “stereotype-name” and/or an icon. For example, it is possible to create a stereotype “SpatialClass” that extends the UML meta-class “Class”. At the model level, this stereotype can be used on classes in UML diagrams to highlight spatial concepts. Tagged values are meta-attributes, *i.e.* they are defined as properties of stereotypes. Finally, a set of constraints should be attached to each stereotype, precisely defining its application semantics to avoid its arbitrary use by designers in models. For example, a constraint can be defined to guarantee that a “SpatialClass” class has a geometric attribute called “geom.”

Proposed UML Profile

In this subsection, we propose a Data-centric Wireless-Sensor UML profile based on the features described in Section 2, which will act as a framework for modelling the data behaviour in WS implemented on Agri-food-oriented ICT applications (e.g. smart farming) or even in different domains.

Our UML profile (Figure 1) is composed by 15 stereotypes (two for Packages, four for Classes, three for Operations and six for Properties), 24 tagged values (six in Classes, five in Properties and 13 in Operations), three data types (enumerations), and a set of constraints. In the first place, we explain the three data types in our profile and how to use them. In the second place, we introduce our UML profile with the description of the central abstract Class stereotype, along with its five general Properties. In the third place, we describe the three implementable Class stereotypes with their three Operations and one specific Property. Moreover, we complement the exposition of these stereotypes with five examples centred in smart-farming applications. Finally, we present the constraints of our profile in three different levels, providing example OCL for each level.

Profile Data Types

The data types in our profile (Figure 1) help to define the tagged values, the three enumerations are:

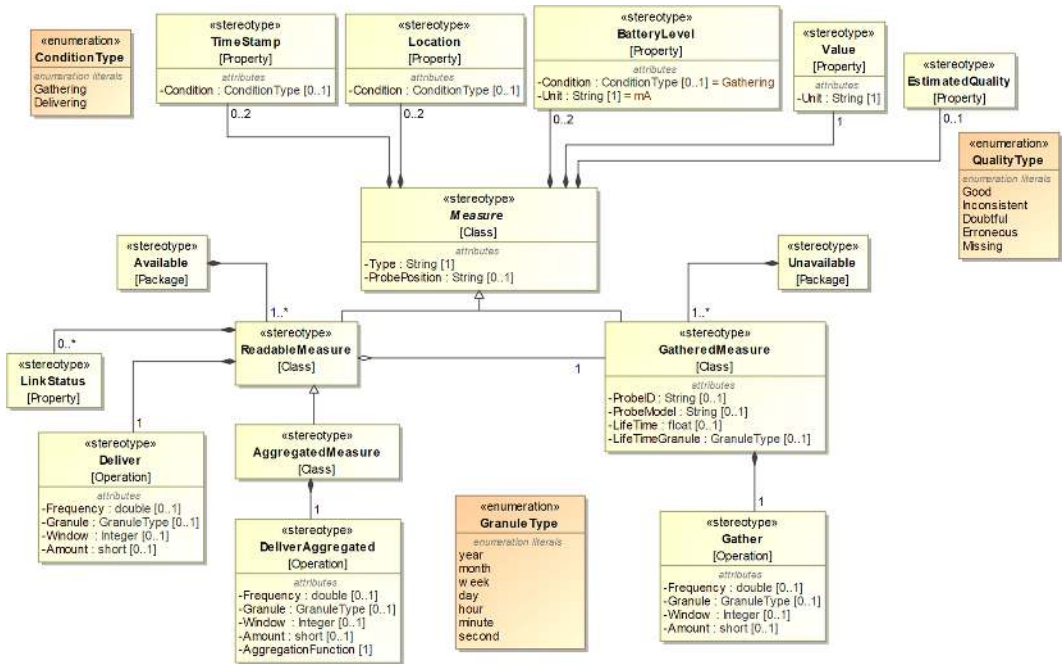
- **ConditionType**: has two possible values (Gathering or Delivering) to indicate in which operation the tagged element was defined.
- **QualityType**: the WSN users, designers or engineers can use the different quality levels to define the how different aspects in their data affects the quality (e.g. battery level or link status) and which is the required dependability of the data. Based on Cantero *et al.* (2016), WSN data can have up to five quality values (these levels are for reference and their full use is not mandatory).
 - Good is the best quality.
 - Inconsistent means that some (few) characteristics of the data indicate a lower quality, but it can be used for non-sensible applications.
 - Doubtful means that the data has low quality and should not be trusted.
 - Erroneous means the data is not good for any application purpose.
 - Missing means there is no data.
- **GranuleType**: defines seven granularities of time that go from second (the smallest granularity) to year (the biggest granularity). This data type is related to the granule tags in the three operations (Gather, Deliver and DeliverAggregated) and the LifeTime.

Profile Abstract Class Stereotype

The main root of our metamodel is the abstract Class Measure, it is intended to identify any measurement gathered, stored or delivered by the WS. The Measure must define a Type (e.g. temperature, humidity, radiation) and could have a ProbePosition (the spatial position of the measuring probe). This Class is composed by five Properties:

- The Value is the main Property for identifying a measurement. It has to be tagged with the measurement Unit.
- The TimeStamp represents a time associated to the measurement. It should have a tagged condition of ConditionType to indicate if it is the time at Gathering or at Delivering the measurement.
- The Location indicates the geometry (the spatial position of the WS) where the measurement is Gathered/Delivered using the ConditionType.
- The BatteryLevel is the remaining energy in the WS at the Gathering/Delivering using the ConditionType. It can be used for triggering low level alerts to indicate that the WS will stop working and the measurement could have lower quality.

Figure 1. Data-centric Wireless-Sensor UML profile from the user point-of-view



- The EstimatedQuality is a derived value that can be calculated in the sensor node in order to estimate the measurement quality. This estimation can consider the remaining energy of the node or the working range of the probe to classify the data in a QualityType.

Profile Implementable Class Stereotypes

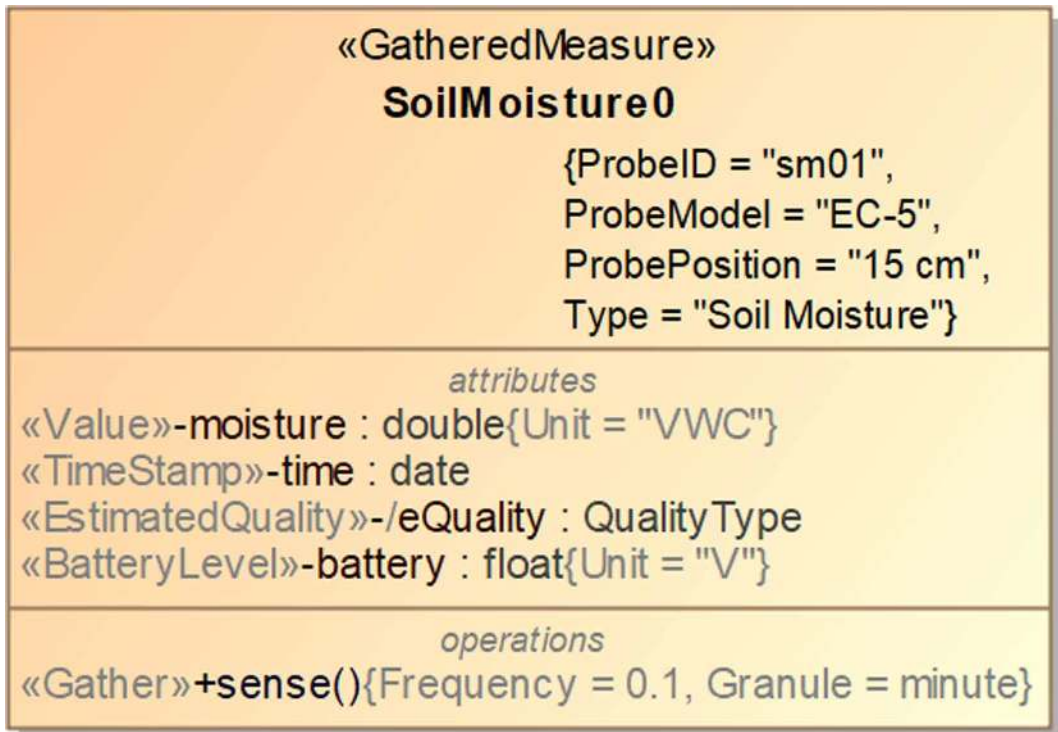
The GatheredMeasure Class is a specification of the abstract Class Measure. It is intended to classify only measurements read through the probes and stored in the sensor node. Consequently, it can be tagged with:

- ProbeID: the identification of the measuring probe,
- ProbeModel: the specific hardware model of the measuring probe,
- LifeTime: the amount of time each measurement will survive in the node,
- LifeTimeGranule: the unit of time for the LifeTime. The time granularities can be from seconds to years, according to the GranuleType.

This Class is composed by one Operation called Gather, which gathers the data from the probe in order to store the measurements. It can be tagged with:

- Frequency: the amount of measurements gathered in a time granule,
- Granule: time unit specifying the Frequency and Window,
- Window: the length (duration) of the Operation’s work cycle in a time granule,
- Amount: maximum number of measurements gathered inside a Window.

Figure 2. Example class implementing the GatheredMeasure stereotype



Finally, as the data of this Class is not available for the application or the user (*i.e.* only exist inside the node), it belongs to the Unavailable Package. Example 1 presents an implementation of this class stereotype.

Example 1

The Class SoilMoisture0 (Figure 2) is an implementation example of the GatheredMeasure stereotype for a sensor node measuring the soil moisture in a crop field. It defines the ProbeID, ProbeModel and Type tags to indicate the node how to process the probe data. Furthermore, the ProbePosition tag allows to describe the measurements by indicating they are gathered from a probe “buried 15 cm into the ground”. The Class attributes show the gathered value is a moisture measured in Volumetric Water Content, and the node must consider the gathering time, the battery level (in Volts) and the estimated quality of each measurement. Finally, the sense operation defines the measurements are gathered at a frequency of 0.1 values per minute (one value each period of 10 minutes).

Table 2 contains an example of the data represented by SoilMoisture0 (Figure 2). This data model allows the sensor node to gather one Moisture measurement (recording Time, Quality and Battery) each 10 minutes.

Moreover, the ReadableMeasure Class is also a specification of the abstract Class Measure, which is intended to classify only measurements sent to the application or the user (*i.e.* available data); thus, it belongs to the Available Package. This Class is composed by one Property and one Operation: LinkStatus and Deliver. The LinkStatus Property is a network connection parameter useful for detecting bad quality in the network connectivity. While the Deliver Operation transmits the stored data to an accessible repository (*e.g.* a database), an application (*e.g.* an information or alert system), or the final user. The tags describing this operation are similar to the tags of the previously described Gather Operation: it can have a delivering Frequency, a Granule, a delivering Window and

Table 2. Example data for SoilMoisture0

Moisture	Time	EQuality	Battery
20	25-10-17 22:03:16	Good	3.7
50	25-10-17 22:13:16	Inconsistent	3.6
21	25-10-17 22:23:16	Good	3.7
21	25-10-17 22:33:16	Good	3.7
13	25-10-17 22:43:16	Inconsistent	3.6

an Amount. Example 2 presents an implementation of this class stereotype; furthermore, Example 3 explains the usage of the GatheredMeasure and the ReadableMeasure in a simple hypothetical case study in smart farming.

Example 2

The Class 3SoilMoisture (Figure 3) is an implementation example of the ReadableMeasure stereotype for a sensor node delivering soil moisture measurements from a crop field. Its definition of ProbePosition and Type comes from the related GatheredMeasure (i.e. SoilMoisture0), indicating a Soil Moisture probe, “buried 15 cm into the ground”, is gathering the measurements. The Class attributes represent data accessible for the application or the final user. These attributes are related to the GatheredMeasure: the sensed-moisture value, the sensed-time timestamp, the estimated quality of the data, and the sensed battery level. This class also defines the sendTime timestamp for the delivery time. Finally, the send operation defines that data should be delivered 0.1 times per minute (once each 10 minutes), but only a maximum amount of three values are delivered inside each 60-minutes window.

Table 3 contains an example of the data represented by 3SoilMoisture (Figure 3). This data model allows the node to deliver one Moisture measurement (including Times, Quality and Battery) each 10 minutes, with a maximum of three measurements per hour. For example, data is delivered during the 22 hour at 22:03; 22:13 and 22:23.

Figure 3. Example Class implementing the ReadableMeasure stereotype

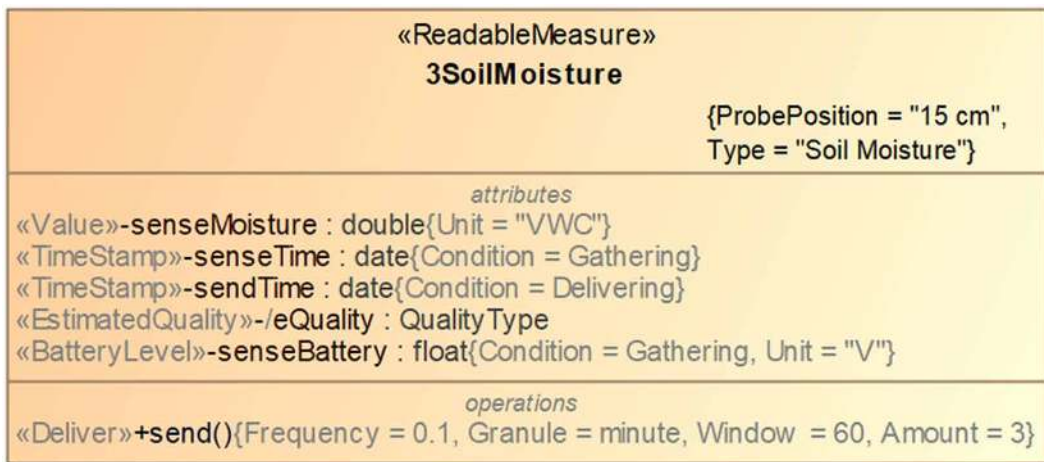


Table 3. Example data for 3SoilMoisture

SenseMoisture	SenseTime	SendTime	EQuality	SenseBattery
20	25-10-17 22:03:16	25-10-17 22:03:16	Good	3.7
30	25-10-17 22:13:16	25-10-17 22:13:16	Inconsistent	3.6
21	25-10-17 22:23:16	25-10-17 22:23:16	Good	3.7
25	25-10-17 23:03:16	25-10-17 23:03:16	Good	3.7
20	25-10-17 23:13:16	25-10-17 23:13:16	Inconsistent	3.6
25	25-10-17 23:23:16	25-10-17 23:23:16	Inconsistent	3.6
14	26-10-17 00:03:16	26-10-17 00:03:16	Inconsistent	3.5

Example 3

These classes (Figure 2 and Figure 3) could represent a single-node application example (Figure 4), on which the hypothetical user (e.g. a farmer) needs to know the soil moisture of the crop field in order to decide if irrigation is needed. The user expects to receive no more than three inconsistent or better-quality information about the soil moisture per hour.

Tables 4 (Gathered) and 5 (Delivered) contain an example of the data represented by this model (Figure 4). These data show that the node gathers one Moisture measurement (recording Time, Quality and Battery) each 10 minutes. Furthermore, it delivers those measurements (including the delivering time) with the same frequency, but only a maximum of three Good- or Inconsistent-quality measurements per hour (Erroneous data is not delivered). For example, among 6 data values collected during the 16 hours, only three values are sent.

In this example, the application designers must define some rules for estimating the quality (e.g. with the battery) and avoiding the delivering of lower-quality data (Example 8). Moreover, they could have defined some rules to stop the WS from gathering data once the delivering operation stops.

Furthermore, the AggregatedMeasure Class is a specification of the ReadableMeasure Class. This Class also identifies available data. However, it is not the data gathered by the probes and stored by the node, it is an aggregate value. Delivering only aggregated data is important since it reduces the network load by transmitting highly meaningful data that enables the applications to work properly with a simple, yet complete, description of the sensed data (Zhang, Han, Cai, & Yin, 2017). Therefore, the AggregatedMeasure Class defines the DeliverAggregated Operation. This Operation is like Deliver, but it includes an additional step: aggregating the stored data inside the window through an AggregationFunction (tagged in the Operation). This allows the WS to make available only highly

Figure 4. UML model of a moisture WS data from the user point-of-view

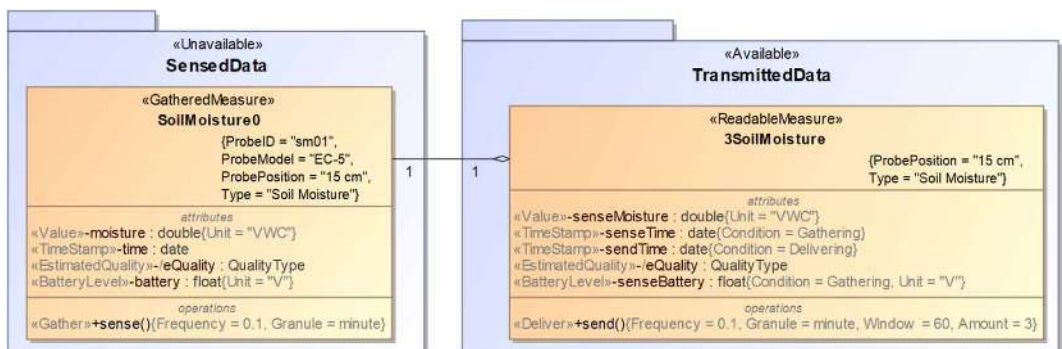


Table 4. Example of the gathered data for the moisture WSN without aggregation

Moisture	Time	EQuality	Battery
...
30	03-12-17 15:45:21	Inconsistent	3.3
30	03-12-17 15:55:21	Inconsistent	3.3
31	03-12-17 16:05:21	Inconsistent	3.3
31	03-12-17 16:15:21	Inconsistent	3.3
30	03-12-17 16:25:21	Erroneous	3.2
31	03-12-17 16:35:21	Inconsistent	3.3
34	03-12-17 16:45:21	Erroneous	3.2
31	03-12-17 16:55:21	Inconsistent	3.3
42	03-12-17 17:05:21	Erroneous	3.2
35	03-12-17 17:15:21	Erroneous	3.2
30	03-12-17 17:25:21	Inconsistent	3.3

useful data. Example 4 presents an implementation of this class stereotype; furthermore, Example 5 explains the usage of the GatheredMeasure and the AggregatedMeasure in a hypothetical case study requiring aggregation in a smart farming application.

Example 4

The Class AggregatedSoilMoisture (Figure 5) implements the AggregatedMeasure stereotype into an example of sensor node delivering aggregated (minimum) soil moisture measurements from a crop field. It defines the ProbePosition and Type tags from a related GatheredMeasure (e.g. SoilMoisture0, though the Class should define a LifeTime to indicate some data persistence), indicating a Soil Moisture probe, “buried 15 cm into the ground”, is gathering the data.

The attributes of AggregatedSoilMoisture (Figure 5) represent data accessible for the application or the final user. These attributes are related to the GatheredMeasure. However, unlike in the 3SoilMoisture example (Figure 3), the delivered data is not the same gathered data. This Class will only deliver, every hour, the minimum moisture measurement, the timestamp of the minimum measurement and the timestamp for the transmission with the sendAgg operation.

Table 6 contains an example of the data represented by AggregatedSoilMoisture (Figure 5). This data model allows the node to deliver the minimum value of the measured Moisture (including the sense and send Time) each hour, since it includes the “Min” aggregation operation.

Example 5

Table 5. Example of the delivered data for the moisture WSN without aggregation

SenseMoisture	SenseTime	SendTime	EQuality	SenseBattery
...
31	03-12-17 16:05:21	03-12-17 16:05:21	Inconsistent	3.3
31	03-12-17 16:15:21	03-12-17 16:15:21	Inconsistent	3.3
31	03-12-17 16:35:21	03-12-17 16:35:21	Inconsistent	3.3
30	03-12-17 17:25:21	03-12-17 17:25:21	Inconsistent	3.3

Figure 5. Example Class implementing the AggregatedMeasure stereotype

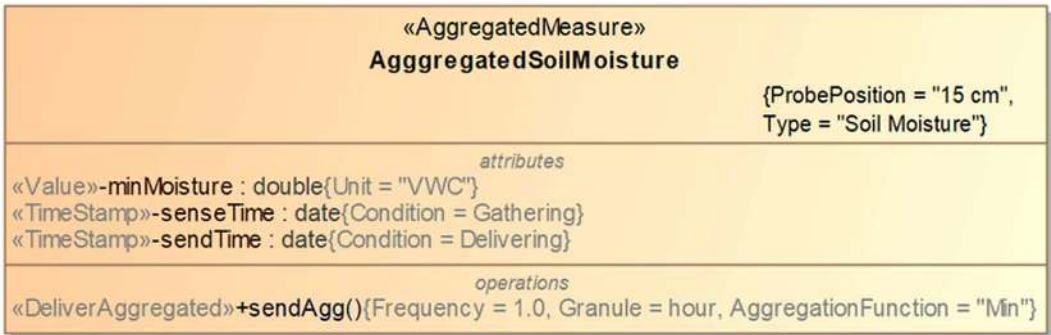


Table 6. Example data for AggregatedSoilMoisture

MinMoisture	SenseTime	SendTime
41	03-12-17 06:20:00	03-12-17 06:59:59
40	03-12-17 07:00:00	03-12-17 07:59:59
38	03-12-17 08:40:00	03-12-17 08:59:59
38	03-12-17 09:10:00	03-12-17 09:59:59

Another application example (Figure 6) could leverage this stereotype: a hypothetical user (e.g. a farmer) needs to know when the soil of the crops is too dry in order to irrigate it. The user expects only good quality information about the minimum soil moisture once per hour. This application (Figure 6) is similar to the first one (Figure 4) with one important difference: the user only requires aggregated data.

This difference implies (as previously stated) the unavailable gathered data in SoilMoisture1 must persist until the aggregation is committed. Therefore, it defines a Lifetime of one hour. Moreover, AggregatedSoilMoisture class provides the user the required information by aggregating the data in SoilMoisture1 each hour with the function “Min” (Minimum) and delivering the aggregated value.

Tables 7 (Gathered) and 8 (Delivered) contain an example of the data represented by this model (Figure 6). These data show that the sensor node gathers one Moisture measurement (recording Time,

Figure 6. UML model of a moisture WS data with aggregation from the user point-of-view

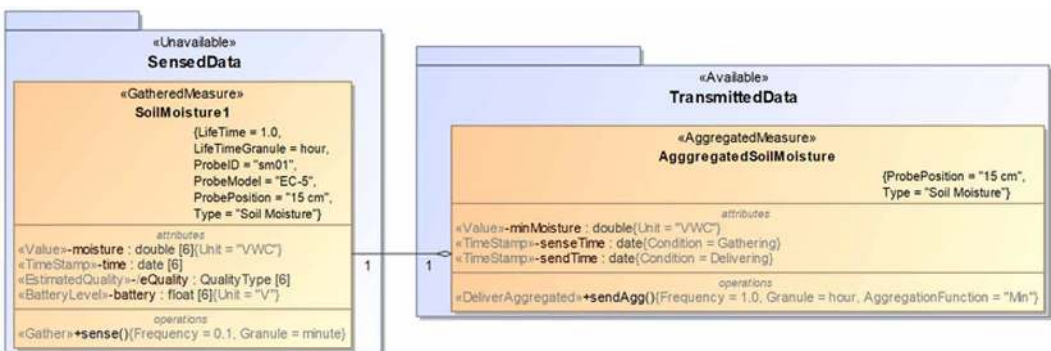


Table 7. Example of the gathered data for the moisture WSN with aggregation

Moisture	Time	EQuality	Battery
...
42	03-12-17 06:50:00	Good	3.5
40	03-12-17 07:00:00	Good	3.5
42	03-12-17 07:10:00	Good	3.5
41	03-12-17 07:20:00	Good	3.5
41	03-12-17 07:30:00	Good	3.5
48	03-12-17 07:40:00	Erroneous	3.4
40	03-12-17 07:50:00	Erroneous	3.4
40	03-12-17 08:00:00	Good	3.5
39	03-12-17 08:10:00	Good	3.5
35	03-12-17 08:20:00	Erroneous	3.4
38	03-12-17 08:30:00	Good	3.5
38	03-12-17 08:40:00	Good	3.5
36	03-12-17 08:50:00	Erroneous	3.4
38	03-12-17 09:00:00	Good	3.5
38	03-12-17 09:10:00	Good	3.5
39	03-12-17 09:20:00	Good	3.5
...

Quality and Battery) each 10 minutes, storing up to six values that last one hour. Furthermore, the WSN delivers an aggregate (minimum) of the gathered moisture values (including the sense and send times) each hour, considering only Good-quality data for the aggregation. For example, among the 6 values gathered at the 7 hour only the one with the minimum value (40) and Good quality is sent.

In this example, the application designers must define some rules for estimating the quality (e.g. with the battery) and avoiding the aggregation of data with non-Good-quality (Example 8).

Profile Constraints

Finally, our Data-centric Wireless-Sensor UML profile also defines a set of constraints, expressed using OCL:

Table 8. Example of the delivered data for the moisture WSN with aggregation

MinMoisture	SenseTime	SendTime
...
41	03-12-17 06:20:00	03-12-17 06:59:59
40	03-12-17 07:00:00	03-12-17 07:59:59
38	03-12-17 08:40:00	03-12-17 08:59:59
38	03-12-17 09:10:00	03-12-17 09:59:59
...

Figure 7. OCL for meta-model level constraints regarding the obligatoriness of a <<Value>> attribute in all the <<Measure>> classes

Context	Measure
Name	Measure-Value attribute
OCL	self.ownedAttribute ->select (m m.ocllsTypeOf(Value))->size(=)1
ErrorMessage	One Value Measure.

- Meta-model level constraints: these constraints are defined at the meta-model level and grant well-formed class diagrams using the UML profile. Example 6 presents two OCL rules of this type.
- Semantic coherence constraints: these constraints are associated to particular elements of our UML profile and they are valid for each application. For example:
 - the Frequency of Delivering (FD) must be equal or less than the Frequency of Gathering (FG);
 - the LifeTime must be equal or greater than the Gathering period (1/FG);
 - the Window (Win) on each operation must be equal or greater than the operation period (1/F);
 - the total amount of stored measurements (Σ SM) cannot be greater than the total node storage (NS);
 - when the Frequency is defined for an operation, the Granule must also be defined for that operation. Moreover, a Window cannot be defined without the Frequency and the Granule. And an Amount requires a Window (besides the Frequency and the Granule);
 - when the LifeTime is defined, the LifeTimeGranule must also be defined, and vice versa. Example 7 implements this rule in OCL.
- *User-defined constraints*: Each model designer, according to the user and application requirements, should define other application-specific constraints, for example deliver only good quality data. Example 8 presents some OCL rules of this type for the hypothetical case studies of Examples 3 and 4.

Example 6

In this example we present one meta-model level OCL constraint. In particular, the rule specifying that any class stereotyped with <<Measure>> (including ReadableMeasure, GatheredMeasure or AggregatedMeasure) must have one (and only one) attribute stereotyped with <<Value>> (Figure 7).

Example 7

In this example we present the OCL for some semantic coherence constraints; in particular for the last two examples: the frequency dependence (Figure 8) and the lifetime granularity (Figure 9).

This constraint (Figure 8) indicates that designers should define at least both the Frequency and the Granule tags for the Deliver operation if they want to use any of the operation tags, including Window and Amount. This constraint can be equally defined for the Gather and DeliverAggregated operations. Nevertheless, note that the AggregationFunction tag is mandatory in the DeliverAggregated operation, regardless the definition of Frequency and Granule.

This constraint (Figure 9) indicates that designers should define both the LifeTime and LifeTimeGranularity tags in the GatheredMeasure class if they want to have persistence in the gathered data.

Example 8

In this example we present some user-defined constraints. Considering the aforementioned application examples (Figure 4 and Figure 6), designers will need to define application-specific constraints in

Figure 8. OCL for semantic-coherence constraints regarding the frequency dependence in the Deliver operation

Context	Deliver
Name	FrequencyDependence
OCL	(Frequency.oclIsUndefined() = Granule.oclIsUndefined()) and (Frequency.oclIsUndefined() implies Window.oclIsUndefined()) and (Window.oclIsUndefined() implies Amount.oclIsUndefined())
ErrorMessage	Frequency and Granule must be defined together or not be defined at all. Also, if Frequency is not defined the Window must not be defined, and if the Window is not defined the Amount must not be defined.

Figure 9. OCL for semantic-coherence constraints regarding the granularity of lifetime in GatheredMeasure

Context	GatheredMeasure
Name	LifeTimeGranularity
OCL	LifeTime.oclIsUndefined() = LifeTimeGranule.oclIsUndefined()
ErrorMessage	LifeTime and LifeTimeGranule must be defined together or not be defined at all.

OCL for each case. The first application (Figure 4) is required to deliver only inconsistent or better data; thus, it needs to identify the quality of the data and reject all the lower-quality values (Figure 10).

The first constraint in Figure 10 is the transmissionStandard, which imposes the delivering of only higher quality data (Good or Inconsistent). Moreover, the second constraint is the qualityStandard, which defines how the battery level affects the data quality in this example application (Good, Inconsistent or Erroneous).

The second application (Figure 6) requires only good-quality data. Thus, it needs to identify the quality of the data and include only good-quality values for aggregation (Figure 11).

The first constraint in Figure 11 is the aggregationStandard, which imposes the aggregation of only Good-quality data. Moreover, the second constraint is the qualityStandard, which defines how the battery level affects the data quality in this example application (Good or Erroneous).

These eight examples illustrate some of the most important stereotypes, tag and constraints of our profile, which allows for a better understanding of its implementation. Furthermore, since

Figure 10. OCL application-specific constraints for example 3

Context	3SoilMoisture
Name	transmissionStandard
OCL	SoilMoisture0->reject(sm sm.eQuality = Erroneous)
ErrorMessage	Send only moisture measures with Good or Inconsistent eQuality.

Context	SoilMoisture0
Name	qualityStandard
OCL	(battery > 3.6 implies eQuality = Good) and ((battery <= 3.6 and battery > 3.3) implies eQuality = Inconsistent) and (battery <= 3.3 implies eQuality = Erroneous)
Error Message	If the battery is above 3.6 V the data quality is good, if it is above 3.3 V and in or below 3.6 V the data quality is inconsistent, but if it is in or below 3.3 V the data quality is erroneous.

Figure 11. OCL application-specific constraints for example 5

Context	AggregatedSoilMoisture
Name	aggregationStandard
OCL	SoilMoisture1->reject(sm sm.eQuality <> Good)
ErrorMessage	Aggregate only moisture measures with Good eQuality.

Context	SoilMoisture1
Name	qualityStandard
OCL	(battery > 3.4 implies eQuality = Good) and (battery <= 3.4 implies eQuality = Erroneous)
Error Message	If the battery is above 3.4 V the data quality is good, but if it is in or below 3.4 V the data quality is erroneous.

the examples 3, 5 and 8 focus on two agriculture-oriented hypothetical case studies, specifically a smart-farming application for irrigation decision support, we can infer that our profile will improve the design phase of this kind of application, easing the meet of the user’s requirements from WSN, including (temporal) data aggregation and early quality assessment.

CASE Tool Implementation

In this subsection, we present the implementation of our UML profile using the CASE tool MagicDraw. MagicDraw is a CASE tool that allows defining UML profiles and OCL constraints defined at class and object levels. Moreover, MagicDraw also allows implementing the meta-model level and semantic coherence constraints. The implementation of our profile provides an automated evaluation of its correctness and consistency (Marouane et al., 2017). Moreover, this implementation allows to use our profile (with stereotypes, tags, and constraints) in the definition of new valid UML models for the WS data behaviour.

An example of the implementation of the OCL constraint of Example 6 (Measure-Value attribute - Figure 7) is shown in Figure 12.

In this example (Figure 12), we define an erroneous element using the ReadableMeasure stereotype. Thus, MagicDraw shows the element that presents the error (*i.e.* humidity class) and the details of the error (*i.e.* Message “one value measure”) according to the defined OCL rule. However, if an element is well-defined, the CASE tool must show nothing.

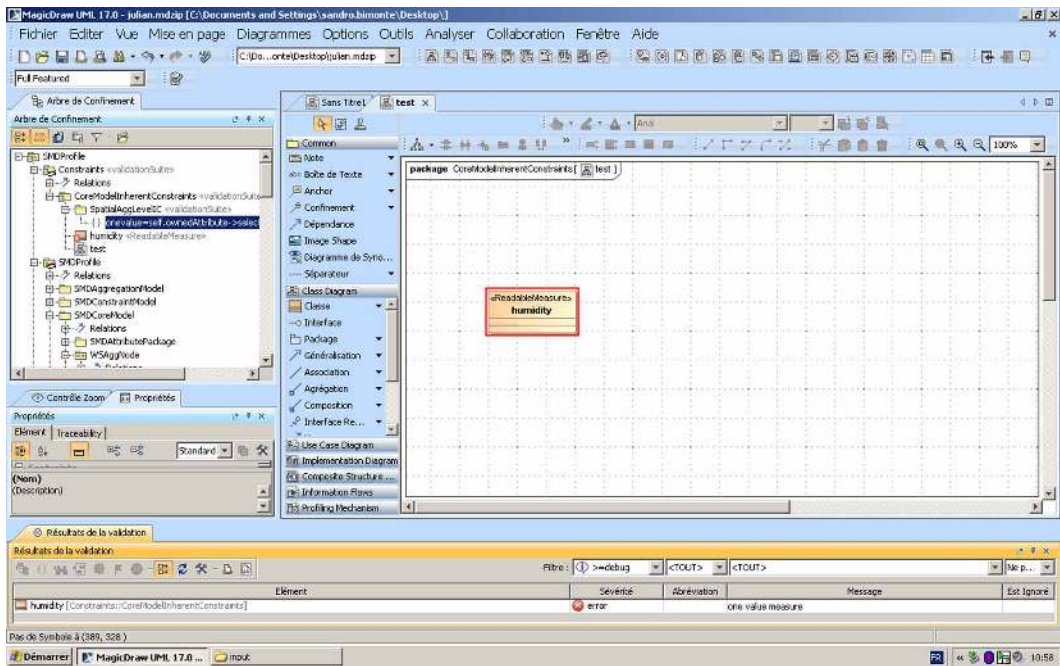
This CASE tool implementation of our UML profile in MagicDraw validates its correctness and consistency with the OCL constraints (Marouane et al., 2017). Hence, we deduce that our profile can be used to design new error-free, smart-farming WSN applications from the WS data, considering the final user’s need.

VALIDATION

In this section, we thoroughly validate our data-centric wireless-sensor UML profile in a real smart-farming case study; therefore, we have modelled the data of the iLive network (Liu, Hou, Shi, & Guo, 2012) of Irstea (French National Research Institute of Science and Technology for Environment and Agriculture).

This iLive network is a result from a partnership between the Irstea institute and the LIMOS (Laboratoire d’Informatique, de Modélisation et d’Optimisation des Systèmes) laboratory. The goals of this experimentation was to evaluate the iLive wireless sensor, developed by the LIMOS, in an agricultural context. The LIMOS went, more precisely, to evaluate energy consumption and fault tolerant capability of their iLive solution for smart-farming applications. The iLive wireless sensors were deployed in the Irstea Montoldre research and experimental site. The description of the iLive data is relevant since this network is part of the projects with others in the topic of robotics that constitute

Figure 12. MagicDraw implementation of our profile with OCL constraints



an initial base for the Irstea AgroTechnoPôle, a project that looks towards the establishment of an innovation ecosystem for the European agricultural industry and academy (Irstea, 2016).

The iLive network is an experimental WSN composed of low-energy devices equipped with a ZigBee wireless communication module, two AA batteries, one air-humidity probe, one air-temperature probe, one light probe (mostly used for laboratory tests), and support for three Decagon (part now of Meter Environment company) probes and four Irrrometer Watermark probes; though not all these latter probes are connected to the nodes. For example, in this experimentation, nodes are only equipped with three Irrrometer Watermark probes. The network consists of one coordinator node and 10 end-devices with a star topology, which are deployed in different fields of the Montoldre site (Figure 13).

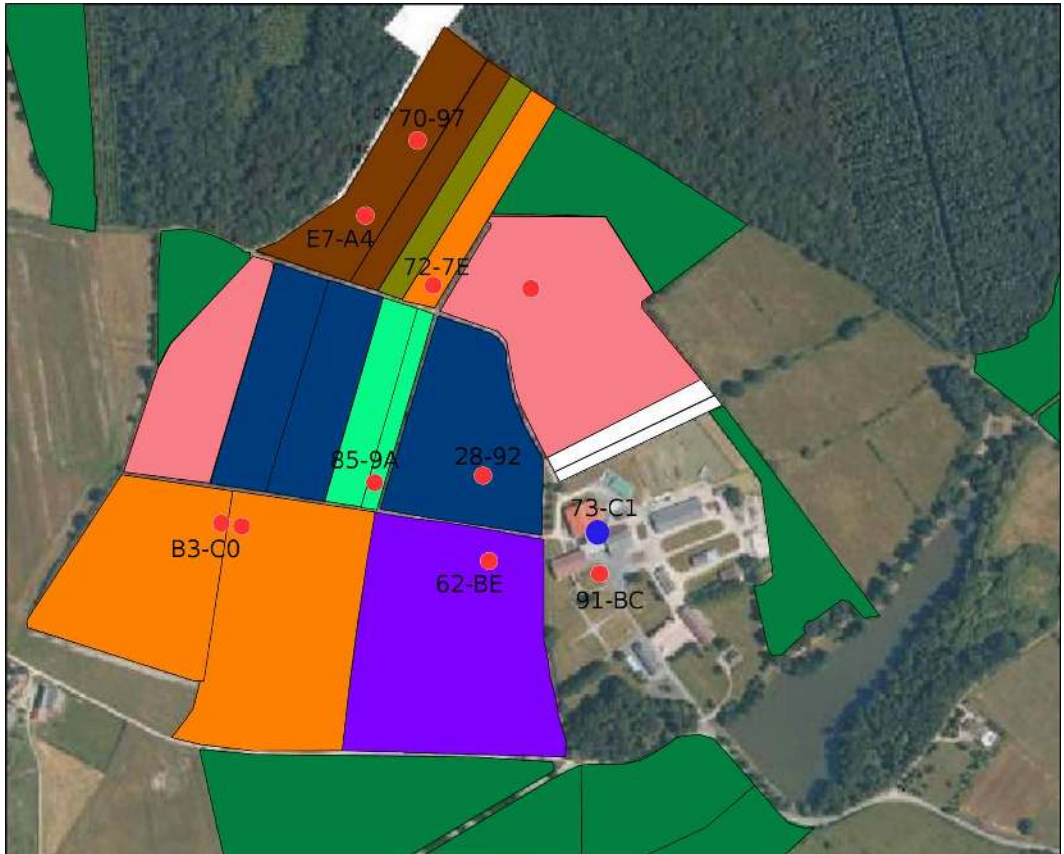
Since the iLive nodes are not equipped with a renewable energy source (e.g. solar panel), they are in Sleep Mode most of the time (about 98%) to reduce energy waste. The nodes work continuously gathering and sending data for about one minute per hour. While the nodes are awake, they gather and deliver data from all their probes 0.111 times per second, which means they make seven measurements per hour.

For modelling and validation purposes, in this paper we analyse a small data subset delivered by one of the iLive nodes: the 91-BC (Table 9).

Based on the analysis of this data (Table 9), the network characteristics, and considering our profile, we propose the following UML model for the description of the data in node 91-BC of the iLive network (Figure 14).

The modelled node (Figure 14) gathers and delivers three types of measurements: Air Humidity in percentage of Relative Humidity (%RH), Air Temperature in degrees Celsius (°C), and Soil Moisture in centibars (cb) or kilopascal (kPa). The measures Air Humidity (Humidity) and Air Temperature (Temperature) are gathered in one irrelevant unknown position. While the Soil Moisture measures (Watermark 1, 2 and 3) are gathered in three relevant known positions (0.3, 0.6 and 1 meters into the ground).

Figure 13. Deployment of the iLive network in the Montoldre site



For every gathered measure, the node delivers the measurement Value, TimeStamp and BatteryLevel. Consequently, all the measured data besides a Link Quality Indicator (LQI) and a Received Signal Strength Indicator (RSSI) for characterizing the link status are delivered to a database to be accessible for the final users.

Moreover, all the gathered measures have the same gathering frequency: 0.111 measurements per second, with a maximum of seven measurements in a 3600 seconds window. This frequency configuration indicates the node will gather measurements each nine seconds, but it will only be working for the first 63 seconds of each hour, collecting a total amount of seven measurements per hour.

Since the iLive nodes send the measurements as soon as they are gathered, the frequency configuration for the deliver operation of the readable measures is the same as the one of the gather one: only seven measurements per hour, delivering each measurement with a nine seconds time span.

Finally, the users can access all the available data (iLSentData). The air temperature and humidity, and different-depth soil moistures allow the farmers to monitor and control their crops. Furthermore, the battery level and link status data allow for technical maintenance of the node and the sensors network, besides the analysis of the data quality.

This model (Figure 14) allows to visualise the data behaviour inside one iLive end-node. Visual models like this one are very important on a system definition, since it allows users, designers, scientists and engineers to check and assess the system feasibility before its implementation. In this particular case, through the analysis of the model (Figure 14), and considering the capabilities of our profile, we infer that the amount of delivered measurements could have been reduced with aggregation functions

Table 9. Data subset for the analysis of the iLive network from the node 91-BC

Node	Humidity	Temperature	Watermark 1	Watermark 2	Watermark 3	Packet Time	Battery	lqi	rssI	dbTime
91-BC	100.00	14.80	30.00	19.00	10.00	6/05/2014 10:01:35	2841	205.00	-83.00	6/05/2014 10:01:35
91-BC	100.00	14.70	30.00	19.00	10.00	6/05/2014 10:01:43	2856	168.00	-83.00	6/05/2014 10:01:43
91-BC	100.00	14.80	30.00	19.00	10.00	6/05/2014 10:01:51	2871	141.00	-83.00	6/05/2014 10:01:51
91-BC	100.00	14.70	30.00	19.00	10.00	6/05/2014 10:02:01	2871	120.00	-83.00	6/05/2014 10:02:01
91-BC	100.00	14.80	30.00	19.00	10.00	6/05/2014 10:02:09	2841	105.00	-83.00	6/05/2014 10:02:09
91-BC	100.00	14.80	30.00	19.00	10.00	6/05/2014 10:02:18	2841	93.00	-83.00	6/05/2014 10:02:19
91-BC	100.00	14.80	30.00	19.00	10.00	6/05/2014 10:02:27	2841	84.00	-83.00	6/05/2014 10:02:27
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:00	2856	205.00	-83.00	6/05/2014 11:04:00
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:08	2841	168.00	-83.00	6/05/2014 11:04:08
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:17	2856	141.00	-83.00	6/05/2014 11:04:17
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:25	2856	120.00	-83.00	6/05/2014 11:04:25
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:33	2841	105.00	-83.00	6/05/2014 11:04:33
91-BC	100.00	13.60	29.00	18.00	10.00	6/05/2014 11:04:42	2841	93.00	-83.00	6/05/2014 11:04:42
91-BC	100.00	13.70	29.00	18.00	10.00	6/05/2014 11:04:50	2856	84.00	-83.00	6/05/2014 11:04:50
91-BC	100.00	15.70	31.00	18.00	10.00	6/05/2014 12:06:25	2841	207.00	-82.00	6/05/2014 12:06:25

like average, which helps to reduce the sensor noise and battery waste in the data transmission (Anisi et al., 2015; Jesus et al., 2017), and the storage requirements of the data-centre. Moreover, the quality of the gathered/delivered data could be estimated from the node from the battery level, link status and the change in the measured values of the same hour.

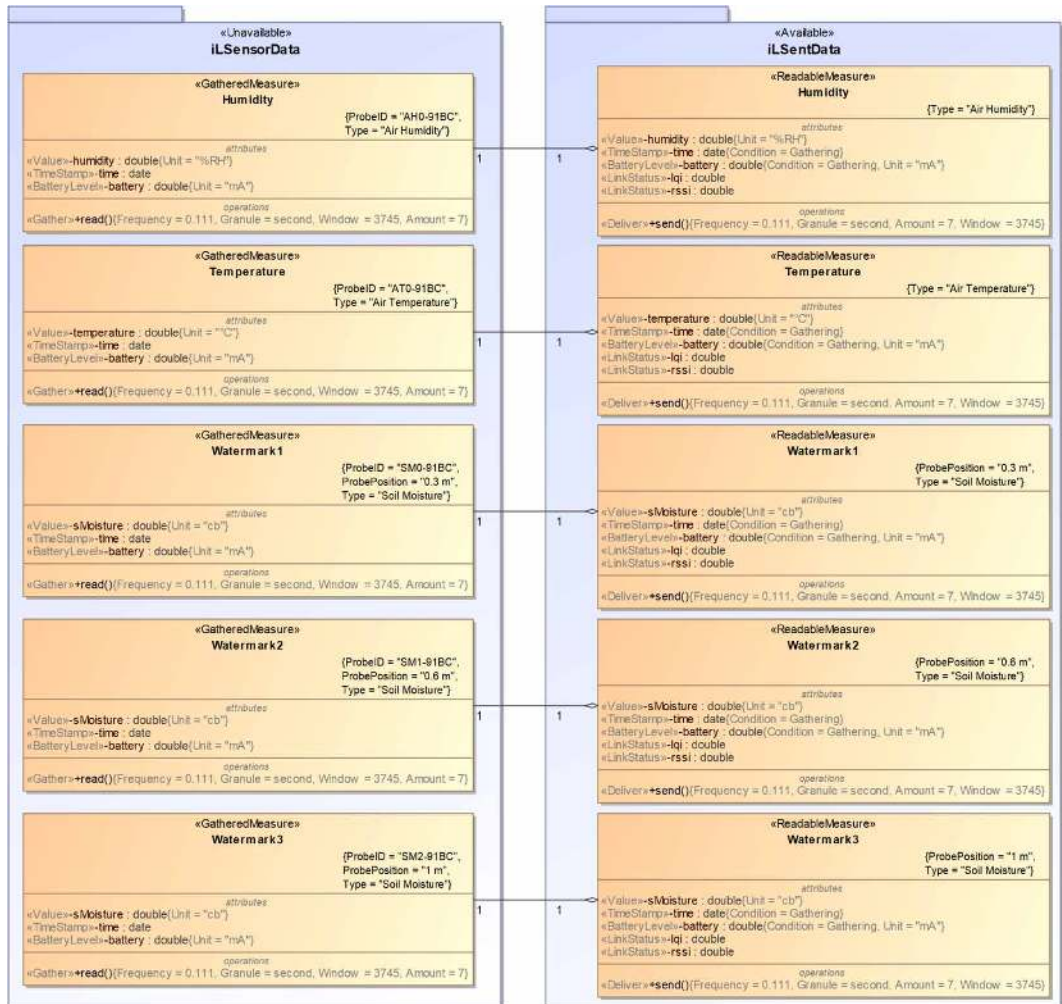
CONCLUSION AND FUTURE WORKS

In this paper, we have presented a UML profile for the design of data collected and managed in wireless sensor nodes from the user point-of-view. Our profile achieves a separation between the gathered (unavailable) data and the delivered (available) data of the node, while describing it with different characteristics and configurations of frequency, aggregation, persistence and quality.

The CASE tool implementation of our profile shows its correctness and consistency. Moreover, the validation on a real smart-farming case study evidences that our profile can be used for the description of data collected by real WS in real WSN applications with different energy-efficient configurations. Besides, the formal (UML) representation of the case study allowed us to conclude that the iLive network designers could have leveraged the aggregation and quality-checking capabilities of our profile in order to reduce the transmission costs and database storage, and to increase the user-perceived value of the available data.

Therefore, this case study illustrates the importance of following a model-driven approach in the design and implementation of WSN applications. Indeed, the conceptual modelling allows for

Figure 14. UML data model from the user point-of-view for the iLive case study, node 91-BC



an abstract and direct analysis of the system properties and behaviour in the design, which could improve the effectiveness and efficiency in the implementation (Abrial, 2010).

When compared with different state-of-the-art approaches, our UML profile lacks of specific analysis methods for evaluating the data quality and dependability in different network levels (Jesus et al., 2017; Prathiba et al., 2016), nor provides complex mechanisms for the execution of multiple data-processing operations in the network (Thang et al., 2011). Nevertheless, our profile sticks to the UML standard to design the data behaviour in the WS from the user point-of-view, with different configurations for the data gathering and delivering, also enabling the temporal aggregation and quality assessment of the data; altogether in a single model.

Different domains could leverage these advantages. For instance, in smart farming our profile could ease and formalise the definition and integration of the sensor-collected data into early warning systems that rely on dependable, aggregated measures (Plazas, Rojas, Corrales, & Corrales, 2016); or machine-learning implementations for the estimation of the crops' yield and meteorological conditions (Plazas et al., 2017; Valencia-Payan & Corrales, 2017).

Hence, our meta-model becomes a first step in driving WSN into a Fog Computing paradigm through a model-driven approach. This change in traditional WSN will allow to improve the value of new Agri-food information systems, since it will reduce the computational and storage load in the central servers, and the communication load in the WSN, providing a faster and more accurate analysis of the monitored environments and extending the network life.

Thereafter, as future works we propose the definition of a joint between the measures that enables the spatial aggregation inside the same node (between probes with the same type of measure); the integration of mechanisms to overtake the memory constraint in some sensor platforms for the unlimited aggregation (distributive and algebraic, not holistic) of temporal data. Furthermore, we also propose an extension of our profile considering the data behaviour in all the WSN levels, including spatio-temporal aggregation mechanisms for inter-nodes data.

ACKNOWLEDGEMENT

We would like to thank Universidad del Cauca and Irstea for supporting our research, the GIT and COPAIN groups for the scientific support, LIMOS and Université Clermont Auvergne (UCA) for sharing the iLive data with us, Innovación Cauca for the mobility scholarship granted to MSc Julián Eduardo Plazas, Colciencias for the PhD scholarship granted to MSc Julián Eduardo Plazas, and Project “Alternativas Innovadoras de Agricultura Inteligente para sistemas productivos agrícolas del departamento del Cauca soportado en entornos de IoT” financed by Convocatoria 04C-2018 “Banco de Proyectos Conjuntos UEES-Sostenibilidad” of Project “Red de formación de talento humano para la innovación social y productiva en el Departamento del Cauca InnovAcción Cauca” ID-4633.

REFERENCES

- Abrial, J.-R. (2010). *Modeling in Event-B: system and software engineering*. Cambridge University Press. doi:10.1017/CBO9781139195881
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4), 393–422. doi:10.1016/S1389-1286(01)00302-4
- Anisi, M. H., Abdul-Salaam, G., & Abdullah, A. H. (2015). A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture. *Precision Agriculture*, 16(2), 216–238. doi:10.1007/s11119-014-9371-8
- Aqeel-ur-Rehman, , Abbasi, A. Z., Islam, N., & Shaikh, Z. A. (2014). A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, 36(2), 263–270. doi:10.1016/j.csi.2011.03.004
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. *Computer Networks*, 54(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010
- Bimonte, S., Schneider, M., & Boussaid, O. (2016). Business Intelligence Indicators: Types, Models and Implementation. *International Journal of Data Warehousing and Mining*, 12(4), 75–98. doi:10.4018/IJDWM.2016100104
- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (pp. 13–16). New York, NY: ACM. doi:10.1145/2342509.2342513
- Cama-Pinto, A., Pineres-Espitia, G., Zamora-Musa, R., Acosta-Coll, M., Caicedo-Ortiz, J., & Sepúlveda-Ojeda, J. (2016). Design of a wireless sensor network for monitoring of flash floods in the city of Barranquilla Colombia. *Ingeniare. Revista Chilena de Ingeniería*, 24(4), 581–599. doi:10.4067/S0718-33052016000400005
- Cantero, F. Y. V., Corrales, J. C., & Gonzalez, G. A. R. (2016). Aproximación a un modelo contextual para calidad de datos en agricultura de precisión. *Revista Ingenierías Universidad de Medellín*, 15(29), 99–112. doi:10.22395/rium.v15n29a6
- Champeau, J., Dhaussy, P., Moitie, R., & Prigent, A. (2000). Object oriented and formal methods for AUV development. In *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings* (Vol. 1, pp. 73–78 vol.1). doi:10.1109/OCEANS.2000.881236
- Firlej, M., & Kresse, W. (2016). Java-library for the access, storage and editing of calibration metadata of optical sensors. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41.
- Gu, G., Zhu, L., Xiong, Z., & Ding, H. (2010). Design of a Distributed Multiaxis Motion Control System Using the IEEE-1394 Bus. *IEEE Transactions on Industrial Electronics*, 57(12), 4209–4218. doi:10.1109/TIE.2010.2044127
- Guillemin, P., & Friess, P. (2009). *Internet of Things: Strategic Research Roadmap*. CERP-IoT. Retrieved from http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf
- Hong, S., Lim, S., & Song, J. (2011). Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks: A Survey. *Transactions on Internet and Information Systems (Seoul)*, 5(4), 805–821. doi:10.3837/tiis.2011.04.010
- Idoudi, N., Duvallet, C., Sadeg, B., Bouaziz, R., & Gargouri, F. (2008). Structural Model of Real-Time Databases: An Illustration. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (pp. 58–65). doi:10.1109/ISORC.2008.80
- Irstea. (2016, February 26). AgroTechnoPôle Auvergne Rhône-Alpes: the creation of an unprecedented structure for innovation in Europe. Retrieved from <http://www.irstea.fr/en/all-news/institute/agrotechnopole-auvergne-rhone-alpes-creation-unprecedented-structure-innovation>
- Jabeen, F., & Nawaz, S. (2015). In-network wireless sensor network query processors: State of the art, challenges and future directions. *Information Fusion*, 25, 1–15. doi:10.1016/j.inffus.2015.01.007
- Jeong, S., Park, H., Lee, J.-E., Lee, S., & Kim, J. (2008). The Study of Navigation Software for GNSS Sensor Station. In *26th International Communications Satellite Systems Conference (ICSSC)*. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2008-5489

- Jesus, G., Casimiro, A., & Oliveira, A. (2017). A Survey on Data Quality for Dependable Monitoring in Wireless Sensor Networks. *Sensors (Basel)*, 17(9), 2010. doi:10.3390/s17092010 PMID:28869505
- Kenchannavar, H., Patkar, G. S., & Kulakarni, U. P. (2010). Modeling of Coverage Preserving Algorithm Using Object Oriented Methodology with UML. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology* (pp. 708–712). New York, NY: ACM. doi:10.1145/1741906.1742071
- Lee, K., & Song, E. (2006). A Wireless Environmental Monitoring System Based on the IEEE 1451.1 Standards. In *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings* (pp. 1931–1936). doi:10.1109/IMTC.2006.328314
- Lehmann, R. J., Reiche, R., & Schiefer, G. (2012). Future internet and the agri-food sector: State-of-the-art in literature and research. *Computers and Electronics in Agriculture*, 89, 158–174. doi:10.1016/j.compag.2012.09.005
- Liu, X., Hou, K. M., Shi, H., & Guo, C. (2012). Efficient middleware for user-friendly wireless sensor network integrated development environment. In *2012 Wireless Advanced (WiAd)* (pp. 22–28). doi:10.1109/WiAd.2012.6296561
- Luvisi, A., Panattoni, A., & Materazzi, A. (2016). RFID temperature sensors for monitoring soil solarization with biodegradable films. *Computers and Electronics in Agriculture*, 123, 135–141. doi:10.1016/j.compag.2016.02.023
- Machado, G. A. de L., Zacaro, P. M. D., & Junior, A. R. de P. (2012). Applying software engineering methodology for designing biomedical software devoted to electronic instrumentation. *Journal of Computational Science*, 8(10), 1601–1614. doi:10.3844/jcssp.2012.1601.1614
- Mae, Y., Takahashi, H., Ohara, K., Takubo, T., & Arai, T. (2011). Component-based robot system design for grasping tasks. *Intelligent Service Robotics*, 4(1), 91–98. doi:10.1007/s11370-010-0072-5
- Marouane, H., Duvallet, C., Makni, A., Bouaziz, R., & Sadeg, B. (2017). An UML profile for representing real-time design patterns. *Journal of King Saud University - Computer and Information Sciences*. doi:10.1016/j.jksuci.2017.06.005
- Marouane, H., Makni, A., Bouaziz, R., Duvallet, C., & Sadeg, B. (2016). Definition of Design Patterns for Advanced Driver Assistance Systems. In *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs* (p. 3:1–3:10). New York, NY: ACM. doi:10.1145/3022636.3022639
- Meyer, S., Sperner, K., Magerkurth, C., & Pasquier, J. (2011). Towards Modeling Real-world Aware Business Processes. In *Proceedings of the Second International Workshop on Web of Things* (p. 8:1–8:6). New York, NY: ACM. doi:10.1145/1993966.1993978
- Mohan, V. M. (2009). *Advancements in Power System Monitoring and Inter-Operability* [PhD Thesis].
- OMG. (2011, March). About the Unified Modeling Language Specification Version 2.4. Retrieved from <http://www.omg.org/spec/UML/2.4/>
- Paulon, A. R., Fröhlich, A. A., Becker, L. B., & Basso, F. P. (2014). Wireless sensor network UML profile to support model-driven development. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)* (pp. 227–232). doi:10.1109/INDIN.2014.6945512
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic Mapping Studies in Software Engineering* (Vol. 8, pp. 68–77). EASE.
- Plazas, J. E., & Corrales, J. C. (2017). Deploying Timely Alerts Through Converged Services: An Application for Colombian Agriculture. In *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change* (pp. 124–138). Cham, Switzerland: Springer. doi:10.1007/978-3-319-70187-5_10
- Plazas, J. E., López, I. D., & Corrales, J. C. (2017). A Tool for Classification of Cacao Production in Colombia Based on Multiple Classifier Systems. In *Computational Science and Its Applications – ICCSA 2017* (pp. 60–69). Cham, Switzerland: Springer; doi:10.1007/978-3-319-62395-5_5
- Plazas, J. E., Rojas, J. S., Corrales, D. C., & Corrales, J. C. (2016). Validation of Coffee Rust Warnings Based on Complex Event Processing. In O. Gervasi, B. Murgante, S. Misra, A. M. A. C. Rocha, C. M. Torre, D. Taniar, ... S. Wang (Eds.), *Computational Science and Its Applications -- ICCSA 2016* (pp. 684–699). Springer International Publishing. doi:10.1007/978-3-319-42089-9_48

- Pozzani, G., & Zimányi, E. (2010). Defining Spatio-Temporal Granularities for Raster Data. In *Data Security and Security Data* (pp. 96–107). Springer; doi:10.1007/978-3-642-25704-9_10
- Prathiba, B., Sankar, K. J., & Sumalatha, V. (2016). Enhancing the data quality in wireless sensor networks #x2014; A review. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)* (pp. 448–454). doi:10.1109/ICACDOT.2016.7877626
- Rajanikanth, K. N., Narahari, Y., Prasad, N. N. S. S. R. K., & Rao, R. S. (2003). A robust and scalable architecture for airborne radar simulation. In *Conference on Convergent Technologies for Asia-Pacific Region TENCON 2003* (Vol. 1, pp. 173–177). doi:10.1109/TENCON.2003.1273308
- Ramirez-Villegas, J., Salazar, M., Jarvis, A., & Navarro-Racines, C. E. (2012). A way forward on adaptation to climate change in Colombian agriculture: Perspectives towards 2050. *Climatic Change*, *115*(3–4), 611–628. doi:10.1007/s10584-012-0500-y
- Shao, Y., Ren, J., & Campbell, J. B. (2018). 9.04 - Multitemporal Remote Sensing Data Analysis for Agricultural Application. In S. Liang (Ed.), *Comprehensive Remote Sensing* (pp. 29–38). Oxford: Elsevier; doi:10.1016/B978-0-12-409548-9.10431-2
- Sharma, A., Lucini, B. A., Stryjak, J., & Kechiche, S. (2015). *Agricultural M2M: a platform for expansion*. GSMA Intelligence. Retrieved from <https://www.gsmainelligence.com/research/2015/03/agricultural-m2m-a-platform-for-expansion/479/>
- Sicari, S., Grieco, L. A., Boggia, G., & Coen-Porisini, A. (2012). DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks. *Journal of Systems and Software*, *85*(1), 152–166. doi:10.1016/j.jss.2011.07.043
- Smart Object Interoperability. (n.d.). Retrieved 4 March 2018, from <https://www.ipso-alliance.org/ipso-community/resources/smart-objects-interoperability/>
- Smuda, W. J., Gerhart, G., Shing, M.-T., & Auguston, M. (2006). A software design approach for heterogeneous systems of unattended sensors, unmanned vehicles, and monitoring stations. In *Unmanned/Unattended Sensors and Sensor Networks III* (p. 639406). International Society for Optics and Photonics; doi:10.1117/12.693035
- Tanik, U. J., & Arkun-Kocadere, S. (2014). Cyber-Physical Systems and STEM Development: NASA Digital Astronaut Project. In *Applied Cyber-Physical Systems* (pp. 33–49). New York, NY: Springer; doi:10.1007/978-1-4614-7336-7_4
- Thang, N. X., Zapf, M., & Geihs, K. (2011). Model Driven Development for Data-centric Sensor Network Applications. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia* (pp. 194–197). New York, NY: ACM. doi:10.1145/2095697.2095733
- Thramboulidis, K., & Christoulakis, F. (2016). UML4IoT—A UML-based approach to exploit IoT in cyber-physical manufacturing systems. *Computers in Industry*, *82*(Suppl. C), 259–272. doi:10.1016/j.compind.2016.05.010
- Uke, S., & Thool, R. (2016). UML Based Modeling for Data Aggregation in Secured Wireless Sensor Network. *Procedia Computer Science*, *78*, 706–713. doi:10.1016/j.procs.2016.02.120
- Valencia-Payan, C., & Corrales, J. C. (2017). A Multiscale Based Rainfall Amount Prediction Using Multiple Classifier System. In *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change* (pp. 16–28). Cham, Switzerland: Springer. doi:10.1007/978-3-319-70187-5_2
- Wang, N., Zhang, N., & Wang, M. (2006). Wireless sensors in agriculture and food industry—Recent development and future perspective. *Computers and Electronics in Agriculture*, *50*(1), 1–14. doi:10.1016/j.compag.2005.09.003
- Wongwirat, O., Paelaong, S., & Homchoo, S. (2009). ET rescue robot. In *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication* (pp. 321–321). doi:10.1109/ROMAN.2009.5326123
- Yu, W., Dong, W. W., & Feng, W. R. (2012). Designing System for Battlefield Environment Simulation by UML. In *2012 International Conference on Industrial Control and Electronics Engineering* (pp. 1977–1980). doi:10.1109/ICICEE.2012.525
- Zhang, K., Han, Q., Cai, Z., & Yin, G. (2017). RiPPAS: A Ring-Based Privacy-Preserving Aggregation Scheme in Wireless Sensor Networks. *Sensors (Basel)*, *17*(2), 300. doi:10.3390/s17020300 PMID:28178197

Zhang, Y., Chen, N., Wei, J., & Huang, T. (2006). Completing UML Model of Component-Based System with Middleware for Performance Evaluation. In *Emerging Directions in Embedded and Ubiquitous Computing* (pp. 72–82). Springer; doi:10.1007/11807964_8

Julian Eduardo Plazas is a PhD student and MSc in Telematics Engineering at Universidad del Cauca, Popayán (Colombia). He has worked in the definition of rural Early Warning Systems through Converged Services and Complex Event Processing for agricultural environments, and in the implementation of Machine-Learning Classifiers for agricultural Decision Support Systems. His current research topics include Wireless Sensor Networks, Machine Learning and Conceptual Modelling for Intelligent Agriculture Systems.

Sandro Bimonte is a researcher at Irstea, and more exactly he is at TSCF. He received his PhD from INSA-Lyon, France (2004–2007). From 2007–2008, he carried out researches at IMAG, France. He is an editorial board member of the International Journal of Decision Support System Technology and the International Journal of Data Mining, Modelling and Management, and a member of the Commission on GeoVisualisation of the International Cartographic Association. His research activities concern spatial data warehouses and spatial OLAP, visual languages, geographic information systems, spatiotemporal databases, and geo-visualisation.

Gil De Sousa is a research engineer (IR) at Irstea (National Research Institute of Science and Technology for Environment and Agriculture) in the research unit TSCF (Technologies and information systems for agricultural systems), in the Copain (Communication and agri-environmental information systems) team. He joined this institute in 2007. His researches also participate to the Irstea AgroTechnoPôle innovation structure. He received a PhD degree in computer science from the Blaise Pascal University, Clermont-Ferrand (France), in 2008. His research interest deals with agricultural and environmental data collection and management. For data collection, Wireless Sensor Networks (WSN), now integrated in the Internet Of Things (IoT) paradigm, are considered. Data management relies on the use of data base and warehouse information systems. Nowadays, his researches focus on energy management on wireless sensor and Quality of Service (QoS) for wireless communication and end user data provision.

Juan Carlos Corrales received the Dipl-Ing and master's degrees in telematics engineering from the University of Cauca, Colombia, in 1999 and 2004 respectively, and the PhD degree in sciences, specialty computer science, from the University of Versailles Saint-Quentin-en-Yve-lines, France, in 2008. Presently, he is a full professor and leads the Telematics Engineering Group (GIT) at the University of Cauca. His research interests focus on big data analytics in agriculture sector.