A Conceptual Model for Query Processing in Wireless Sensors: an Application to the Agriculture

Julián Eduardo Plazas Pemberthy

Master's Thesis in Telematics Engineering

Supervisors:
Dr Sandro Bimonte, PhD
Dr Gil de Sousa, PhD
Dr Juan Carlos Corrales Muñoz, PhD

Universidad Del Cauca
Faculty of Electronics and Telecommunications Engineering
Department of Telematics
Field of e-Agriculture
Popayán, August 2018

Julián Eduardo Plazas Pemberthy

A Conceptual Model for Query Processing in Wireless Sensors: an Application to the Agriculture

Dissertation submitted to the
Faculty of Electronics and Telecommunications Engineering
of the Universidad del Cauca, Colombia
for the acquisition of the academic degree

Magíster en:
Ingeniería Telemática

Supervisors:
Dr Sandro Bimonte, PhD
Dr Gil de Sousa, PhD
Dr Juan Carlos Corrales Muñoz, PhD

Popayán
2018

Universidad del Cauca

Universidad del Cauca
Faculty of Electronics and Telecommunications Engineering
Postgraduate Programs

FORM I
DEFENSE ACT OF A
MASTER'S DISSERTATION / DOCTORAL THESIS

The jury of the:

MASTER'S DISSERTATION (**X**)   DOCTORAL THESIS ( )

Entitled:
A Conceptual Model for Query Processing in Wireless Sensors: an Application to the Agriculture

Under the supervision of: **Dr Juan Carlos Corrales Muñoz, Dr Sandro Bimonte, and Dr Gil de Sousa**

HEREBY STATE:

That being the **8:00** hours of **August 28, 2018**, the student:

Julián Eduardo Plazas Pemberthy

accomplished the defense of the master's dissertation / doctoral thesis, obtaining the grade:

APPROVED (✓)   ~~REJECTED ( )~~.

In witness whereof, the present Act is signed on **August 28, 2018**.

JUROR 1 (or Program Director*):

François PINET

Name: Dr François Pinet

JUROR 2 (Doctorate):

Name:

COORDINATOR JUROR:

Name: Dr Gustavo Adolfo Ramírez González

(*) The program director sign with the authorization of the jury:

Este trabajo está dedicado especialmente a mi familia
Muchas gracias por su apoyo y cariño constantes
Por alentarme a dar siempre lo mejor de mí

Acknowledgements

I would like to thank doctors Juan Carlos, Sandro and Gil for making me a better researcher with their tutoring and advice. Also, I want to thank Universidad del Cauca, AgroCloud project of RICCLISA and Irstea for supporting this research, the GIT and COPAIN groups for their scientific and personal support, the LIMOS laboratory, and the Aquarisc and Silkworms Incubator projects for sharing their data and comments, Innovacción Cauca for my mobility scholarship and Colciencias for my PhD scholarship.

Finally, I must thank my family and friends for making this process more pleasant, and thanks be to God for making all this possible.

Structured abstract

Background: The Agri-food sector, along with other application domains, obtain high benefits from the monitoring capacity of Wireless Sensor Networks (WSN). This kind of applications should supply the users' needs while integrated with different information systems. Such requirements could be seamlessly achieved in a model-driven approach through a complete and effective conceptual design phase. Current model-driven approaches allow to model WSN applications considering different communication and processing configurations. However, to the best of our knowledge, they do not allow for a complete and clear description of the node's data and the aggregation operations applied to them.

Aims: The main aim of this thesis is to define a conceptual model for processing aggregation queries inside individual wireless sensor nodes of agriculture-oriented WSN. Modelling WSN data behaviour is relevant since it would allow to evaluate the capacity of an application for supplying the end-user needs, increase the cooperation capacity between the system's designers, engineers, scientists and users, and define a transparent integration with different data-centric information systems.

Methods: To achieve this aim, this thesis was divided in four steps. In the first place, a systematic mapping with review allowed to describe the problem domain for an agricultural application. In the second place, a systematic literature review allowed to characterize the main wireless sensors used in agriculture. In the third place, a conceptual modelling process allowed to define an abstract representation of the wireless sensors' data behaviour in UML, including the description of the data gathering, aggregation and delivering operations. Finally, in the fourth place, a thorough validation process with academics, case studies and CASE-Tool testing allowed to prove the feasibility of the proposed model.

Results: Through this process, we have defined a list with some of the most relevant challenges, characteristics and constraints in the design of agriculture-oriented WSN applications. From these features, we have built a conceptual meta-model in UML, which allows for the design of data querying in wireless sensor platforms, supporting the execution of temporal aggregation operations. Moreover, in this meta-model, we have achieved an explicit representation of the implicit separation between the data that is unavailable for the end-users and the data that is available and is used by the different information systems.

Conclusions: Our UML profile provides a standardized, complete and effective representation of the wireless sensors' data behaviour from the user point-of-view, which could allow for the implementation of model-driven WSN applications that supply the end-user needs. Indeed, the models generated with our UML profile help visualise the ideal data behaviour in wireless-sensors-based systems, specifying their structure and operations, and allowing for the implementation of the real wireless sensors and their documentation. Moreover, since our UML profile provides different features for configuring the node operations, aggregating the gathered data, and checking the data quality, it allows increasing the user-perceived value of the WSN.

Keywords: Wireless Sensor Networks, UML Profile, Data-Centric, Data Modelling, Model-Driven, Aggregation

Resumen estructurado

Antecedentes: En varios dominios, como en el sector agrícola y alimentario, las capacidades de monitorización de las redes de sensores inalámbricas (WSN, del inglés *Wireless Sensor Networks*) se pueden aprovechar de muchas maneras. Este tipo de aplicaciones debe poder conectarse a diversos sistemas de información para lograr suplir las necesidades de los usuarios finales. Una aproximación orientada por modelos permitiría cumplir fácilmente este objetivo a través de una fase de diseño exhaustiva, completa y efectiva. Las aproximaciones existentes permiten modelar las aplicaciones de WSN considerando distintas configuraciones en comunicación y computación; sin embargo, éstas no ofrecen mecanismos que permitan hacer una descripción clara y completa de los datos monitorizados y las operaciones de agregación que se les aplica a los mismos.

Objetivos: El objetivo principal de esta tesis es la definición de un modelo conceptual del procesamiento de consultas de agregación en nodos sensores inalámbricos individuales de WSN orientadas a la agricultura. El modelado del comportamiento de los datos en WSN es importante ya que permitiría evaluar la capacidad de una aplicación para suplir las necesidades de los usuarios finales, incrementaría la capacidad de cooperación entre los diseñadores, ingenieros, científicos y usuarios de la WSN, y ayudaría a la definición de mecanismos centrados en datos para la integración transparente de distintos sistemas de información.

Métodos: Para alcanzar este objetivo, esta tesis se dividió en cuatro fases: En primer lugar, un mapeo sistemático con revisión permitió describir las principales características, retos y limitantes de la implementación de WSN para aplicaciones agrícolas. En segundo lugar, una revisión sistemática de la literatura permitió caracterizar los sensores inalámbricos que más se utilizan para la agricultura. En tercer lugar, un proceso de abstracción y modelado conceptual permitió la definición de una representación formal en UML del comportamiento de los datos en los sensores inalámbricos, incluyendo las operaciones de obtención, agregación y envío de datos. Finalmente, un triple proceso de validación en distintos ámbitos académicos, con estudios de caso, y con pruebas a través de una herramienta CASE permitió probar la factibilidad del modelo propuesto.

Resultados: A través de este proceso, hemos extraído una lista con algunos de los desafíos, características y limitantes más relevantes en el diseño de WSN orientadas a la agricultura. A partir de estas características, hemos construido un meta-modelo conceptual en UML (perfil) que permite modelar las consultas de datos en sensores inalámbricos, soportando también la definición de operaciones de agregación temporal. Además, con este perfil hemos logrado hacer una representación explicita de la separación que existe implícitamente entre los datos que son recolectados por los sensores (y no están disponibles por fuera de éstos), y los datos que los sensores envían a los usuarios o a distintos sistemas de información para su análisis como un soporte para la toma de decisiones.

Conclusiones: Este perfil UML provee una representación estandarizada, completa y efectiva del comportamiento de los datos en sensores inalámbricos desde el punto de vista del usuario. Esta representación podría permitir la implementación de WSN dirigidas por modelos capaces de cumplir con los requerimientos de las aplicaciones y los usuarios finales. De hecho, los modelos generados con este perfil UML ayudan a visualizar el comportamiento ideal de los datos en sistemas basados en sensores inalámbricos, especificando su estructura y operaciones, y facilitando la implementación y documentación de las WSN reales. Además, dado que este perfil UML provee distintas funcionalidades para la configuración de las operaciones en los nodos sensores (incluyendo la agregación de los datos recolectados y la estimación de su calidad), éste permite incrementar el valor percibido por los usuarios de las WSN.

Palabras Clave: Redes de Sensores Inalámbricos, Perfil UML, Centrado en Datos, Modelado de Datos, Dirección por Modelos, Agregación

Contents

Figure list

Table list

Acronym list

CASE: Computer-Aided Software Engineering
ICT: Information and Communication Technologies
IoT: Internet of Things
IPSO: Internet-Protocol Smart Objects
JSON: JavaScript Object Notation
OCL: Object Constraint Language
UML: Unified Modelling Language
WS: Wireless Sensor nodes
WSN: Wireless Sensor Networks

Chapter 1

Introduction

1.1. Statement of the Problem

Nowadays, most information systems rely on the collection, organisation, and integration of different types of data to provide complete analysis for decision support, situation management, control, etc. Thereby, high-quality sensors data is a common requirement of these kinds of systems [1].

Indeed, the advent of low-cost sensors enabled the development of small sensing platforms with 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 [2]. 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 [3]–[5].

However, handling this kind of data is challenging since the monitoring sensors can collect and stream large amounts of raw data (*e.g.* embedded in different systems) and must deal with limited and depletable resources (*e.g*. deployed in fields of difficult access) [6], [7]. Thus, different applications must correctly and timely process these big data heterogeneous streams in order to improve the decision-making, control, and definition of strategies on their domains (*e.g.* agriculture or environmental monitoring), considering the end-user needs. For instance, WSN data processing and analysis is crucial to handle complex agricultural applications, such as phenology monitoring, yield estimation, or environmental risk [8]. Moreover, the deployment of such a 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 been widely accepted as a relevant technique to build solutions for real complex tasks apart from the implementation problems and limitations [9].

Different researchers have provided important advances for the definition of monitoring-relaying applications involving WSN through conceptual modelling [10]–[15]. In this context, the Unified Modelling Language (UML) is one of the most powerful tools for formalising conceptual models, a widespread extensible object-oriented standard that closes the gaps between designers, developers, and final users [16]. However, to the best of our knowledge, current approaches do not provide a complete and effective conceptual representation of Wireless Sensor node (WS) data in a standardised and simple way, which makes difficult to design complex monitoring-relying applications and reduces the applications' capacity to completely supply the end-user needs [17], [18]. Moreover, from a data-centric approach, queries in WS are responsible for meeting the user/application requirements of information, since they define which data is sensed and how is it processed before reaching the users.

Considering this scenario, we propose to answer the following research question through this master's thesis:

How to design the query processing in individual Wireless Sensors?

1.2. Motivation

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. In Colombia, only the agricultural sector represents more than 10% of the National Domestic Product and the livelihood of almost 4 million people [3], [19].

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 like precision agriculture, smart farming, or product tracking are vital for overcoming these challenges [3], [5], [19], [20].

Such innovations rely on the intensive monitoring of the products (*e.g.* crops) and their environments, and accurate information and forecasting systems that allow for decision support, situation management, control, etc. Thus, a model-driven approach could enhance the design and implementation of these kinds of systems in the Agri-food domain, which would increase their efficiency and effectiveness.

Thereafter, in this master's thesis, we present a data-centric UML profile for agricultural 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 various user-defined queries. Moreover, among the different sensor computation methods, we focus on data aggregation in this thesis. It is necessary for saving the battery lifetime of WSN, useful for complex applications, and the aggregated information can be used to make faster and more accurate decisions [21].

1.3. Objectives

1.3.1. General Objective

To define a conceptual model for processing aggregation queries inside individual wireless sensor nodes of agriculture-oriented WSN.

1.3.2. Specifics Objectives

- To describe the problem domain for an agricultural application.
- To characterise some wireless sensors used in the problem domain.
- To model the query processing in the wireless sensors, enabling support for data aggregation queries.
- To evaluate the feasibility of the proposed model.

1.4. Contributions

The main contributions of this master's thesis are:

- A list of some of the most relevant challenges, characteristics and constraints in the design of agriculture-oriented WSN applications - Chapter 3.
- A conceptual model implemented as a UML profile (with stereotypes, tags, constraints, and data types), which allows for the design of data querying (including aggregation operations) in wireless sensor platforms - Chapter 4.
- An explicit representation of the implicit separation between the data gathered by the node (unavailable for the user/application) and the data delivered by the node (available for the user/application) - Chapter 4.
- An accurate visual description of the data querying in WS of three different WSN applications, which allow for a better understanding of the sensors and data behaviour - Chapter 5.
- One paper relating the main outcomes of this thesis ([22]), approved for publication in the International Journal of Agricultural and Environmental Information Systems (IJAEIS), Volume 10 - Issue 2; indexed in the Web of

Science ® (ESCI) and Scopus ® (SJR Q4). In this dissertation, we present all the outcomes from the original paper, explaining them in-depth. Besides, a pre-copy-edited version of the paper is available in Appendix B.

- The beginning of an alliance between the Universidad del Cauca (Colombia) and the Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture (France) for the development of joint research projects.

1.5. Contents of the Dissertation

This dissertation is divided into six chapters, which we describe as follows:

Chapter 2. State of the Art:
In this chapter, we present a brief description of the most relevant concepts for this thesis. Moreover, we also analyse different research projects working on sensor data, aggregation, and/or conceptual modelling to establish our research gaps.

Chapter 3. Features Discovery:
In this chapter, we study current literature regarding data aggregation, WSN, and agriculture in order to extract a list with some of the most relevant challenges, characteristics and constraints of aggregation in agricultural WSN applications, which will allow us to define our conceptual model.

Chapter 4. Conceptual Model:
In this chapter, we present the process to build the meta-model for the design of the query processing in WS, considering aggregation. This process includes the characteristics selection, grouping, relation, and formalisation. We present the final conceptual model as a UML profile implemented in a Computer-Aided Software Engineering (CASE) tool, which allows for the definition of UML models to design the (aggregation) queries in WS.

Chapter 5. Validation:

In this chapter, we present the validation of our conceptual model in three different ways: firstly, we validate our conceptual model in two academic frameworks with experts in WSN and conceptual modelling from Irstea, the Université Clermont Auvergne (UCA), and the Universidad del Cauca. Secondly, we use our UML profile to model three different case studies of WSN applications on agriculture, one from Irstea and two from Universidad del Cauca. Finally, we use the selected CASE tool to evaluate the correctness and consistency of our conceptual model.

Chapter 6. Conclusions and Future Works:

Finally, from the analysis of our conceptual model and its validation process, we obtain the most important conclusions and propose different future works that could increase the impact of our meta-model.

Chapter 2

State of the art

In this chapter, we introduce the theoretical and technological base for this thesis, analysing the concepts of WSN, query processing, aggregation, and conceptual modelling. Moreover, we present a bibliographic study from which we select and analyse the most relevant related works for this thesis, establishing research gaps and work focuses.

2.1. Background

With the aim of offering a general context and technological background for this thesis, we present a brief description of the main concepts surrounding the definition of a conceptual model for the design of aggregation queries in wireless sensors.

2.1.1. Wireless Sensor Networks

A sensor can be any device capable of representing physical world conditions as measured data [23]. The bases 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 in 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. In 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 [2].

Different WSN application domains, like smart agriculture, require deploying 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 with some quality-checking techniques for the data reliability [23].

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. Besides, resources associated with 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 [24]. 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 [25]. A synthetic definition of the IoT concept is as follows [26]: "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 on the deployment strategy and cartography.

For the effects of this thesis, we only focus on the data collection and management considerations, since their design could allow to early assess the effectiveness and efficiency of a WSN application (*e.g*. for agriculture).

2.1.2. Aggregation

Data aggregation is the process of summarising a set of data through statistical analysis, obtaining a new small, highly-valuable set of data with more descriptive attributes (*e.g.* minimum, maximum, average, standard deviation, mode). In the context of precision agriculture and smart farming, various research projects use aggregated data to feed different information systems, since it reduces the processing time while increasing their precision [4], [16], [27], [28].

Furthermore, early data aggregation is important in WSN for saving resources (*e.g.* nodes battery and central storage) and analysing relevant events occurring in different spatial and temporal scales sooner than in the central servers [29].

2.1.3. Query Processing

Query processing can be considered as the capacity of a data management system (*e.g.* a database) to extract information on-demand, executing some operations over the requested data. In the context of WSN, query processing can be considered as the ability of the network to deliver the user-requested data. However, since the sensor nodes are programmed in a laboratory before their deployment, their capacity of answering queries is limited to their (fixed) definition of the data behaviour, *i.e.* the query processing in WSN is defined before the implementation [30].

Therefore, in this thesis, we focus on the modelling of the data behaviour in the WS (*i.e.* how the nodes gather, process, and deliver the sensed data). Through this kind of model, the network users should be able to understand (with some knowledge in the modelling language) how the data is processed by the nodes and what kind of data they receive. Moreover, network engineers and designers could define new WS implementations considering the user's data requirements, and thus capable of answering the users' queries.

2.1.4. Conceptual Modelling

A conceptual model is an abstract, partial and simplified representation of a system under study, which may or may not exist in the real world (*e.g.* for the design of new systems in a model-driven approach), often used to analyse and understand the system without considering it directly [31]. Models allow sharing a common vision of the system under study among technical (*e.g.* designers and developers) and non-technical parties (*e.g.* final users). Such common vision facilitates the communication between the parties, supporting a more effective and efficient design, development, and maintenance of complex systems, and allowing for a more objective and precise project control [16], [31].

However, a model itself can be considered as a system, with its own identity, elements, relations, and complexity. Therefore, when considering a model is the system under study, the concept of meta-model appears: a model for modelling (designing, describing, and building) models. Moreover, since models must be defined in a modelling language, a meta-model can also be defined as "a model that defines the structure of a modelling language" [31].

Therefore, conceptual modelling can be considered as the conscious, qualitative process of representing a complex system under study as a construction of co-related concepts, variables, and factors, synthesized in a conceptual model [32]. According to Jabareen ([32]), this process is composed of eight phases:

1. Select the most important data sources about the system under study.
2. Categorise the data sources by area and importance.
3. List all concepts, variables, and factors present in the data sources.
4. Deconstruct each element to understand it.
5. Integrate and group the listed elements.
6. Synthesise the elements into a theoretical model (knowledge in the field is very important in this phase).
7. Validate the model in presentations, seminars, conferences and publications to receive feedback from other academics and practitioners.
8. Rethink the conceptual model. There will always be space for improving a conceptual model, especially in highly dynamic fields.

2.2. Related works

Reducing the computational load on central servers that process and analyse big data streams produced by (wireless) sensor networks in Agri-food applications could allow for a 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 to reduce the storage and computing overload in the central servers by delivering only highly-valuable data [6], [7], [33]. This initial analysis can be achieved through different kinds of data aggregation.

Therefore, through a systematic mapping study based on [34], 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 focus on identifying five kinds of aggregation scopes:

- Temporal aggregation: when aggregation is realized in time (temporal) windows over data in a fixed geographical position, usually from one single node.

- Spatial aggregation: when the aggregation is performed to reduce the amount of data produced at the same time in different geographical (spatial) positions, usually from sets of sensor nodes.
- Spatial and Temporal: when spatial and temporal aggregations are performed.
- OLAP (OnLine Analytical Processing) (*i.e.* statistical) aggregation: when sensor data can be aggregated in multiple dimensions (*e.g.* spatial, temporal, and thematic) considering different granularities.
- Query aggregation: when the network reduces the signalling load by delivering more data on fewer messages. This case is irrelevant for this thesis since this scope does not allow for data analysis.

Furthermore, the aggregation is performed at 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 this thesis 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 at 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 signalling in the network. This case is irrelevant for this thesis since this level is not used for analysing the data.

The resulting map (Figure 2.1) represents the relationship between the aggregation scopes and where it is performed in the WSN hierarchy, showing in blue bubbles the research projects relating a distributed processing approach and in yellow bubbles the projects relating a central processing approach. This map evidences the OLAP-like aggregation is only used in the central servers in a centralised approach; besides, the research projects focusing on the analysis of the data through aggregation inside the nodes (WS) are scarce. Furthermore, we have not found any research working on data aggregation only in the temporal dimension, which could be a very good option for analysing the sensed data inside the WS before delivering it to the next network level and to the central server.

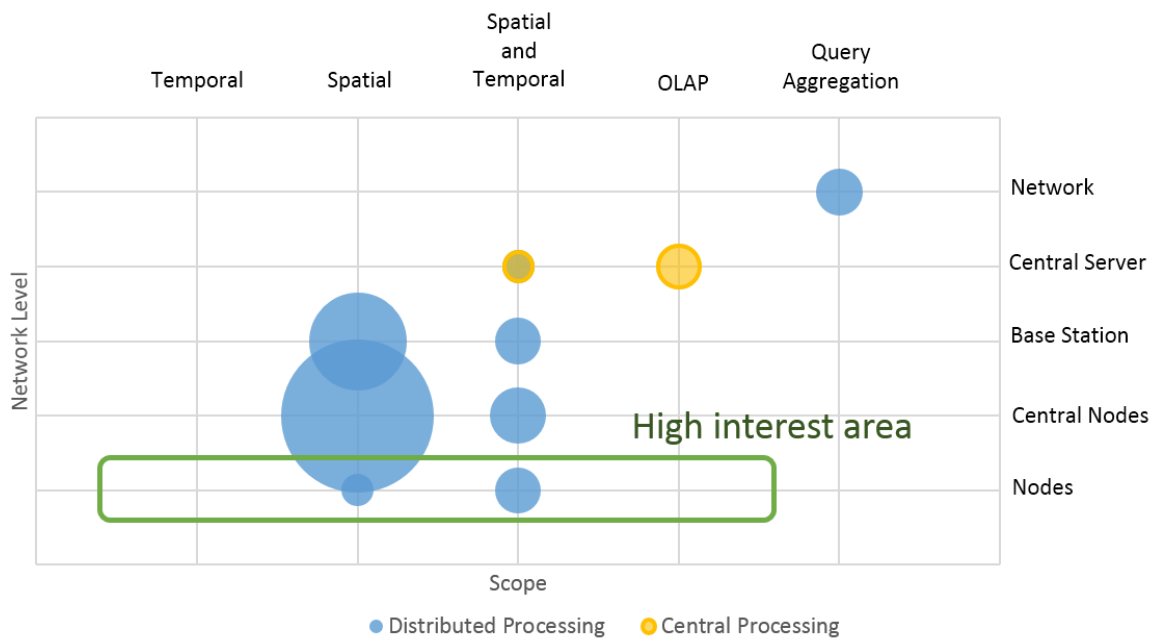

Figure 2.1. Map of researches relating aggregation in WSN.

Considering the results of our study (Figure 2.1), in this thesis, we focus on the more elementary aggregation and its location implementation: temporal aggregation in the nodes (WS). Indeed, in a hybrid decentralized approach like Fog Computing, this type of aggregation should be the basis for the definition of more complex, multi-dimensional types of aggregation in higher network levels. Allowing to reduce the computational load on central servers and enabling a faster data processing and analysis [33].

Moreover, the WSN data must meet the user and application requirements for a successful implementation. An accurate design in a model-driven process supported with conceptual meta-models (*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 [34] 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 2.1). Our study considered the following classification criteria:

1. Genericity: 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. The considered levels are:
    - Simulation: the presented model describes a software tool for simulating WSN (sensors), sensor-related systems (service), or another kind of system.
    - Service: the presented model describes a service that could have a relation with a WSN, without focusing on such a relationship.
    - Network: the presented model describes some network configuration parameters of the WSN (*e.g.* topology or inter-node communication).
    - Physical: the presented model describes some physical features of the WSN or the sensor nodes (*e.g.* intra-node communication or resource management).
    - Application: the presented model describes an application based on WSN, making a special focus on the WSN issues for the application.
    - Data: the presented model describes a WSN application, a WSN, or a sensor node from the collected, received, processed, delivered or stored data.
4. UML profile: the conceptual model is represented as a UML profile with stereotypes, tagged values, and constraints defined in Object Constraint Language (OCL).
5. CASE tool implementation: the conceptual model is implemented in a CASE tool.
6. Sensors implementation: the conceptual model is implemented over existing sensor nodes.

Table 2.1. Classification of the identified research

Paper	Genericity	Meta-model	Design level	UML profile	CASE tool Implementation	Sensors implementation
[10]	Application-specific Driver Assistance Systems	Yes	Physical Application	Yes	Yes MagicDraw	No
[35]	Application-specific Industrial Systems	Yes	Physical Network Application	Yes	Yes Papyrus	No
[36]	Application-specific Early Warning Systems	No	Service	No	No	Yes
[37]	Application-specific Driver Assistance Systems	No	Data	No	Yes MagicDraw	No
[38]	Application-specific Agriculture	No	Application	No	Yes	Yes
[11]	General	No	Sensors Simulation	No	Yes StarUML	No
[12]	General	Yes	Data	No	No	No
[39]	General	No	Application	No	Yes Enterprise Architect	No
[15]	General	Yes	Application Data	No	Yes	No
[40]	Application-specific Education	No	Service	No	No	No
[41]	Application-specific Medicine	Yes	Service	No	Yes Enterprise Architect	Yes

[42]	Application-specific Military	No	Service Simulation	No	Yes StarUML	No
[43]	General	No	Physical	No	Yes	No
[44]	General	No	Application	No	No	No
[45]	General	No	Application Service	No	No	No
[46]	Application-specific Robotics	No	Service Physical	No	No	Yes
[47]	General	No	Application	No	No	Yes
[48]	Application-specific Visual Surveillance Systems	No	Service	No	Yes Visual Paradigm	No
[49]	Application-specific Robotics	No	Service	No	Yes	Yes
[50]	Application-specific Satellite Navigation Systems	No	Service	No	Yes	No
[51]	General	Yes	Physical	No	Yes Enterprise Architect	No
[52]	General	Yes	Physical Service	Yes	Yes	No
[53]	General	Yes	Network	No	Yes Generic Modelling Environment	No
[54]	General	No	Network	No	Yes Rhapsody	Yes
[55]	General	Yes	Physical Network	No	Yes	No
[56]	Application-specific Military	No	Simulation	No	No	No
[57]	General	Yes	Physical	No	Yes Argo	No

The results for this study (Table 2.1) show that most research projects 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.

Such results (Table 2.1) evidence that formal standardised models for describing the sensors' data and applications are scarce. 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 following:

In the first place, Marouane *et al.* ([37]) use UML to represent structural and behavioural information in sensor nodes for an Advanced Driver Assistance System (ADAS) application in order 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.* ([10]) propose an evolution of their previous work ([37]) 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 standardised language.

In the third place, Thramboulidis and Christoulakis ([35]) provide a UML profile for OMA LWM2M (Lightweight machine-to-machine communication protocol) and standard Internet-Protocol Smart Objects (IPSO) in IoT. 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.* ([12]) 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, Nguyen *et al.* ([15]) propose a DSL (Domain-Specific Language) 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.

On the other hand, in this study we also found two relevant secondary studies:

- Malavolta and Muccini ([17]) study the current (2014) approaches on Model-Driven Engineering for WSN in a systematic literature review. The study compares and describes 16 different conceptual models for sensors and WSN, discussing the main issues and challenges for modelling WSN and their applications.
- Essaadi *et al.* ([18]) provide an overview about the state of the art regarding WSN modelling (2017). The paper focuses on the existing modelling languages for the different aspects of WSN (e.g. communication, hardware, software, data, etc.), describing, classifying, and evaluating more than 20 different modelling languages. From this analysis, the study proposes various recommendations for the definition of new models, meta-models, and modelling languages for WSN-based systems.

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 standardised, simple, formal, interoperable, 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 is not supported by existing works [58].

This constitutes a significant gap for model-driven processes of design and implementation of agricultural information systems based on distributed-processing WSN aiming to meet user-related and application-specific requirements.

Summary

In this chapter, we have described the most relevant concepts and technologies for the development and understanding of this thesis:

- WSN, an interconnection of smart devices with sensing capabilities (sensor platforms) and wireless interfaces.
- Aggregation, the process of statistically analysing the data, and a strategy for increasing the WSN lifetime and reducing the computing overload in the central servers.
- Query processing, the capacity for answering the user’s data requirements.
- Conceptual modelling, the process for building an abstract, partial and simplified representation of a system, which allows for its study apart from the implementation problems and limitations.

Moreover, through the analysis of the related works, we have identified research gaps in the execution of temporal aggregation operations inside WS, and on the modelling of the data processed by WS. Indeed, we have found no evidence of formal, standardised, conceptual meta-models that allow modelling the data behaviour in individual sensor nodes considering their modularity (available and unavailable information), enabling the execution of aggregation operations, and supporting quality and dependability characteristics in a single unified model.

Chapter 3

Features Discovery

The first step in the definition of a conceptual meta-model for the design of the query processing in individual sensor nodes is the discovery and description of the features that allow describing the sensors and their data (aggregated or not). Thus, based on the systematic literature review [59], we conduct a tertiary study on sensors, sensor data, and aggregation in the agricultural context. This study allows us to identify, understand, and classify such features for the definition of our conceptual model.

In this chapter, we present the results of our literature review ([59]) process. In the first place, we list and categorise the most relevant agriculture-oriented sensors. In the second place, we state the most relevant issues of WSN considering their implementation in the agricultural context and the execution of aggregation operations inside the network. Finally, we identify the features that allow for describing the WS and their data, considering the analysis of the listed sensors and challenges.

3.1. Agricultural Sensors

In this section, we analyse, select and list the most relevant agriculture-oriented sensors based on the studies of Aqeel-ur-Rehman *et al.* ([23]) and Jawad *et al.* ([21]).

Also, we classify the sensors considering two parameters: what they monitor (Monitoring), and the type of the sensor (Type).

The first parameter (Monitoring) relates the three main aspects to consider in the monitoring of agricultural crops [23]:

- Soil: Is the ground where the crops are grown; thus, its nutrients and water contents are important to have good products.
- Plant/Leaf: Monitoring some crops' plants is important to check the development of the products and for estimating the harvest yield.
- Environment: Different meteorological conditions affect the crops in more than one way, *e.g.* an optimal temperature range will allow for optimal plant growth [60], and combinations of various conditions allow for the appearance of diseases [27].

The second parameter (Type) considers the hardware characteristics of the physical sensor:

- Probe: These are the most basic sensors. They leverage the physicochemical properties of some material to measure their surroundings, delivering constant analogue responses (usually voltage). Nevertheless, some advanced proves can deliver digital measurements (Section 2.1).
- Integrated Probe: This is a more advanced sensor that measures more than one property of its surroundings by gathering different probes in one. Usually, these integrated probes deliver digital measurements through a single (wired) communication channel.
- Advanced Sensor: These sensors do not rely only on physicochemical changes to measure their surroundings. Indeed, they have physical adaptations in special conditions that allow them to calculate a condition (*e.g.* rain) by measuring simple changes (*e.g.* displacement).
- Sensor Platform: This is a low-cost hardware platform with some processing capabilities that support different probes to measure its surroundings. This type of sensor can be connected to the energy or rely on batteries.
- Wireless Sensor Platform: This is a subtype of sensor platform that integrates a wireless module and usually relies on batteries as the energy source. The main purpose of these sensors is to be part of a WSN as nodes.

- Station: This is an advanced sensor platform that has increased memory and computing capabilities, supports advanced sensors, and relies on multiple energy sources (*e.g.* batteries and solar panels).
- Portable Station: This is a lightweight hardware platform with some integrated sensors that can measure some parameters of its surroundings. These portable stations are designed to be carried by a person who takes the measurements in different places.

Table 3.1 organises the identified sensors in six columns: the sensor name, the link to the sensor web page or its datasheet, the sensor provider (manufacturer or merchant), the sensor's classification in the first parameter (Monitoring), the sensor's classification in the second parameter (Type), and the different variables that the sensor can measure (for the Stations and Sensor Platforms, this last column indicates the supported measures).

Table 3.1. Agriculture-oriented sensors
Source: [21], [23]

Sensor	Link	Provider	Monitoring	Type	Measures
Hydra probe II soil sensor	http://www.stevenswater.com/products/sensors/soil/hydraprobe/	Stevens Water Monitoring Systems	Soil Related	Integrated Probe	Real dielectric permittivity (isolated); Soil moisture; Bulk electrical conductivity; Temperature; Inter-sensor variability
MP406	http://www.ictinternational.com/products/mp406/mp406-moisture-sensor/	ICT International	Soil Related	Integrated Probe	Volumetric water content (Soil moisture); Soil water potential
Pogo portable soil sensor	http://pogoturfpro.com/pogo/pro/	Stevens Water Monitoring Systems	Soil Related	Portable Station	Moisture; Salinity (EC); Canopy temperature; Salinity index
ECH2O EC-5	https://www.decagon.com/en/soils/volumetric-water-content-sensors/ec-5-lowest-cost-vwc/	Decagon Devices	Soil Related	Probe	Volumetric water content (Soil moisture)

ECRN-50 low-REC	https://www.decagon.com/en/canopy/canopy-environment/ecrn-50-low-resolution-rain-gauge/	Decagon Devices	Soil Related	Advanced Sensor	Rain; Water volume per hour
WET-2	http://dynamax.com/products/soil-moisture/wet-2-water-conductivity-temperature-sensor	Dynamax	Soil Related	Integrated Probe	Water content; Electrical conductivity; Temperature
VH-400	http://www.vegetronix.com/Products/VH400/	Vegetronix	Soil Related	Probe	Dielectric constant of the soil (Soil moisture)
THERM200	http://www.vegetronix.com/Products/THERM200/	Vegetronix	Soil Related	Probe	Temperature
Tipping bucket rain gage	http://www.stevenswater.com/products/sensors/meteorology/tipping-bucket/	Stevens Water Monitoring Systems	Soil Related	Advanced Sensor	Rain
AquaTrak 5000	http://www.stevenswater.com/products/sensors/hydrology/level/aquatrak/	Stevens Water Monitoring Systems	Soil Related	Advanced Sensor	Absolute liquid level
ECRN-100 high-REC rain Gauge	https://www.decagon.com/en/canopy/canopy-environment/ecrn-100-high-resolution-rain-gauge/	Decagon Devices	Soil Related	Probe	Rain
BetaTherm 100K6A1B	http://uk.farnell.com/betatherm/100k6a1b/thermistor-ntc/dp/9707220	--	Soil Related and Environment Related	Probe	Temperature
107-L temperature sensor	https://www.campbellsci.es/107	Campbell Scientific	Soil Related and Leaf/Plant Related	Probe	Temperature
237-L, leaf wetness sensor	https://www.campbellsci.cc/237-l	Campbell Scientific	Leaf/Plant Related	Probe	Leaf Wetness (electrical resistance on the surface)

Leaf wetness sensor	https://www.decagon.com/en/canopy/canopy-measurements/lws-leaf-wetness-sensor/	Decagon Devices	Leaf/Plant Related	Probe	Leaf Wetness; Ice Formation
LW100, leaf wetness sensor	http://www.globalw.com/products/lw100.html	Global Water	Leaf/Plant Related	Probe	Leaf Wetness; Rain
SenseH2™ hydrogen sensor	https://www.ntmsensors.com/products/hydrogen-sensors/ntm-senseh2-hydrogen-sensor/	NTM Sensors	Leaf/Plant Related	Probe	Hydrogen (H2) concentration in air
LT-2 M (leaf temperature)	http://www.solfranc.com/productos/?wpcproduct=sensor-de-temperatura-de-la-hoja-lt-2m&lang=en	Solfranc Tecnologias	Leaf/Plant Related	Probe	Leaf temperature
TPS-2 portable photosynthesis	http://ppsystems.com/tps2-portable-photosynthesis-system/	PP Systems	Leaf/Plant Related	Portable Station	CO2; H2O
PTM-48A photosynthesis monitor	http://phyto-sensor.com/PTM-48A	Phyto-Sensor Group	Leaf/Plant Related	Station	CO2 & H2O exchange rates
YSI 6025 chlorophyll sensor	https://www.ysi.com/Accessory/id-6025/6025-Chlorophyll-Sensor	YSI	Leaf/Plant Related	Probe	Chlorophyll in water
Field scout CM1000TM	https://www.spectrumeters.com/nutrient-management/chlorophyll-meters/chlorophyll/cm1000/	Spectrum Technologies	Leaf/Plant Related	Portable Station	Relative chlorophyll index
CI-340 hand-held photosynthesis	http://www.solfranc.com/productos/?wpcproduct=ci-340-equipo-portatil-de-fotosintesis	Solfranc Tecnologias	Leaf/Plant Related and Environment Related	Portable Station	CO2; H2O; Air temperature; Light intensity

Met station one	http://www.stevenswater.com/resources/documentation/msoquickref.pdf	Stevens Water Monitoring Systems	Environment Related	Station	Wind Speed; Wind Direction; Temperature; Relative humidity; Barometric Pressure
CM-100 compact weather station	http://www.overtech.ntv.com.br/files/datasheet/file/369/cm-100datasheet.pdf	Stevens Water Monitoring Systems	Environment Related	Station	Wind Speed; Wind Direction; Temperature; Relative humidity; Barometric pressure
CS300-L Pyranometer	https://www.campbellsci.cc/cs300-pyranometer	Campbell Scientific	Environment Related	Probe	Total solar radiation
HMP45C	https://www.campbellsci.cc/hmp45c-l	Campbell Scientific	Environment Related	Integrated Probe	Temperature; Relative humidity
SHT71	https://www.sensirion.com/en/environmental-sensors/humidity-sensors/pintype-digital-humidity-sensors/	Sensirion	Environment Related	Integrated Probe	Temperature; Relative humidity
LI-200 Pyranometer	http://www.stevenswater.com/products/sensors/meteorology/li200/	Stevens Water Monitoring Systems	Environment Related	Integrated Probe	Global Solar Radiation
XFAM-115KPASR	https://www.pewatron.com/en/products/pressure-sensors-load-sensors/product/xfam/	Fujikura	Environment Related	Probe	Absolute Pressure
SHT75	https://www.sensirion.com/en/environmental-sensors/humidity-sensors/pintype-digital-humidity-sensors/	Sensirion	Environment Related	Integrated Probe	Temperature; Relative humidity
Met One Series 380 rain gauge	http://metone.com/meteorological-sensors-systems/rain/370-380/	Met One Instruments	Environment Related	Advanced Sensor	Rain

WXT520 compact weather station	https://www.campbellsci.com/wxt520	Campbell Scientific	Environment Related	Station	Temperature; Relative humidity; Wind speed; Wind direction; Barometric pressure; Rain
All-In-One (AIO) Weather Sensor	http://www.climatronics.com/Products/Weather-Station-Systems/AIO_compact_weather_station.php	Climatronics	Environment Related	Station	Temperature; Relative humidity; Wind speed; Wind direction; Barometric pressure
RM Young (model 5103)	http://www.youngusa.com/products/7/5.html	R. M. Young	Environment Related	Advanced Sensor	Wind speed; Wind direction
RG13/RG13H	https://www.vaisala.com/sites/default/files/documents/RG13%28H%29-Datasheet-B010195EN-G.pdf	Vaisala	Environment Related	Advanced Sensor	Rain
SHT11	https://www.sparkfun.com/datasheets/Sensors/SHT1x_datasheet.pdf	Sensirion	Environment Related	Probe	Temperature; Relative humidity
MICA2DOT	https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf	Crossbow Technology	Environment Related	Wireless Sensor Platform	Light; Temperature; Accelerometer
MICA2	https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf	Crossbow Technology	Environment Related	Wireless Sensor Platform	Light; Temperature; Humidity; Barometric pressure; Accelerometer; GPS; RH; Acoustic; Video sensor; Microphone; Sounder; Magnetometer
Imote2	http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf	Crossbow Technology	Environment Related	Wireless Sensor Platform	Light; Temperature; Humidity; Accelerometer
TelosB	http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf	MEMSIC	Environment Related	Sensor Platform	Light; Temperature; Humidity

Waspmote	http://www.libelium.com/products/waspmote/	Libelium	Soil, Leaf/Plant, and Environment Related	Wireless Sensor Platform	Leaf wetness; Soil moisture; Fruit diameter; Solar radiation; Humidity; Temperature; Wind speed; Wind direction; Rain

From the identified sensors, the Wireless Sensor Platforms are the only ones designed to be deployed as nodes in a WSN (*i.e.* a WS), which enables the monitoring of large agricultural fields and the gathering of massive amounts of crop-related data for the implementation of precision agriculture and smart-farming processes [21], [61]. Hence, the Waspmote, MICA2DOT, MICA2 and Imote2 are some of the most relevant sensors for this thesis.

3.2. WSN Issues

In this section, we leverage the results of our study to identify relevant issues of implementing WSN in agricultural applications (Table 3.2) and of executing aggregation operations inside these kinds of networks (Table 3.3). Besides, we classify each issue considering the aspects of WSN they are related (Relation). The analysis of these issues is important for this thesis since it will allow us to discover relevant features for the description of WS data.

For our classification scheme (Relation), we define four aspects of WSN which the issues can challenge:

- Data: the issue is directly related to the gathered, processed or delivered data of the WSN, *i.e.* the information about the crop conditions that enables the decision support.
- Software: the issue is related to the logical operation and configuration of the sensor nodes, *i.e.* their internal programming defining how they work as individuals and as part of a network.

- Hardware: the issue is related to the physical constitution, operation and configuration of the sensor nodes, *e.g.* their components, batteries, memory, in-field distribution, etc.
- Telecommunications: this issue is directly related to the data transmission in the WSN, including node-to-node, node-to-station and multi-node communications.

In the first place, based on the works of Peres *et al.* ([61]), Aqeel-ur-Rehman *et al.* ([23]), and Jawad *et al.* ([21]), we state the main issues challenging the implementation of WSN for agricultural applications (Table 3.2). These issues are important for our conceptual-modelling process since the agricultural context presents specific challenges and requirements to WSN, despite most of the identified WS are designed for their deployment on hard environments (*e.g.* industrial or open field).

Table 3.2. Issues for implementing agricultural WSN

Issue	Description	Relation	Reference
Energy Consumption	The nodes are usually equipped with limited and finite energy sources (*e.g.* batteries); thus, the network lifetime greatly depends upon the batteries lifetime. Although replacing the batteries of agricultural WSN is possible, executing this action more than twice a year can be tedious in large, remote fields. Therefore, this issue could be solved in more than one way. Firstly, a proper energy management strategy in hardware and software to reduce the energy consumption of the node functions, especially the data transmission. Secondly, energy-harvesting techniques like solar cells to reduce the batteries usage; however, these techniques highly increase the cost of the network.	Software; Hardware	[21], [23], [61]
Communication Range	Most agriculture-oriented WSN suffer from the effect of harsh conditions because of the open agricultural surroundings; thus, they rely on robust but short communication range protocols. Therefore, to effectively cover large fields, many sensor and router nodes are deployed in the WSN extending its range. However, this increases the cost of the network and the energy consumption due to the re-transmitted data.	Telecommunications	[21]

Propagation Losses	Besides the diverse agricultural surroundings, topographies and climate conditions, agriculture-oriented WSN must communicate through a heavy crop canopy. These circumstances degrade the link quality by causing signal propagation absorption, reflection, attenuation, and scattering. Therefore, this issue must be solved by implementing robust protocols and correct channel path planning, which should ensure the Quality of Service of the network while keeping the energy consumption low.	Telecommunications	[21]
Routing	In large agricultural WSN, different routing problems like packet collision may emerge due to the limited bandwidth and the channel propagation. Therefore, different effective multi-hop transmission strategies must be adopted to avoid these problems.	Telecommunications	[21]
Location and Tracking	Since agricultural WSN can be used to locate and track livestock, several considerations, such as radio interference, animal situation and mobility, changes in the network topology, penetration depth of the signal through the animal body, height of the collar, and access point antennas, need to be taken into account.	Software; Hardware; Telecommunications	[21]
Reliability	The information gathered by the WSN is important for improving the decision making through precision agriculture and smart farming. Thus, this information must be trustworthy and reliable to provide accurate decision support. Moreover, risky situations must be immediately reported and handled in an emergency, which means that the data transmission in WSN must be reliable.	Data; Telecommunications	[21], [58]
Scalability	In agricultural WSN applications, the construction of effective, fault-tolerant and robust networks requires the deployment of several sensor and router nodes in the fields. Therefore, these large-scale WSN must rely on hierarchical, scalable architectures. Moreover, these nodes could generate massive amounts of raw data that do not necessarily yield proportional amounts of information. Thus, the raw data should be aggregated inside the network to allow for scalability, and to reduce the energy and storage waste.	Data; Software; Hardware; Telecommunications	[21], [61]
Cost	Since the agricultural WSN usually deploy several nodes, their software and hardware costs should be very low without sacrificing their performance. This issue should be addressed from the design of the nodes, lowering their costs even more.	Software; Hardware	[21]

Real Time	Since the agricultural WSN data support the decision making in precision agriculture and smart farming processes, the monitoring and reporting of the different crop situations must be opportune. Indeed, real time allows for better management of emergency situations like fires.	Data; Software	[21]
Data Storage	Due to the several sensor nodes deployed in WSN for large-scale agricultural applications, these networks generate huge amounts of crop-related data. Therefore, the storage and analysis centres require a high capacity to record and process all the information, though this requirement could be reduced by aggregating the data in the nodes.	Data; Hardware	[21]
Security	The security of the crop fields is an important issue since insects, rodents or even thieves might attack the farms. Thus, protection and security could be achieved through real-time analysis of different factors without relying on human intrusion.	Data; Software; Hardware	[21]
Delay Tolerance	Although most agricultural applications of WSN require real-time processing of the gathered information, some applications like fire detection or pests exposure are delay-sensitive. In this case, the information must be transmitted as fast as possible to handle the critical issue despite the high energy waste. Therefore, different energy management strategies should be defined in the nodes according to the different events monitored and their delay tolerance.	Software; Telecommunications	[21]
Fault Tolerance	Sensor nodes deployed in open, harsh environments are prone to physical damage, blockage, and interference, which affect the quality of the gathered and delivered data. Therefore, different techniques should be implemented to maintain the reliability of the networks, so the failure of a sensor node should not affect the overall task of the network. For example, data aggregation and topology control schemes could increase the fault tolerance and reliability of a network.	Data; Software; Hardware; Telecommunications	[21], [23]
Data Management	The data management in agriculture poses a challenge since the several sensors deployed in the agricultural fields collect large amounts of data of different natures. Indeed, the designers must define different analysis methods, collection schemes, and sensors types before managing the data. Hence, different techniques like IoT integration or big data should be implemented for the design, deployment, and management of the network.	Data; Software	[21], [61]

Heterogeneity	The task of getting meaningful information from many disparate sensors is not trivial, and the nature of the data also brings specifics difficulties. Indeed, WSN usually integrate various kinds of sensor nodes, measuring different variables with different frequencies, configurations, and requirements. Thus, frameworks, models, or middlewares are important for reducing the high variability in the data.	Data; Software	[21], [23], [61]
Data Acquisition, Processing and Transmission	These are the three main processes of the sensor nodes, and each one of them consumes different amounts of energy. Therefore, these processes should be defined independently in the nodes considering different configurations (*e.g.* frequency), which would allow supplying the applications' requirements without sacrificing the batteries' lifetime.	Data; Software	[23], [61]
Node Size and Housing	Since several sensor nodes are deployed in large fields, they must be small enough to be easily deployable. Moreover, the nodes need a protective housing to resist harsh environmental factors like heat, cold, and rain, and physical mishandling by human or animals. Finally, these characteristics cannot affect the nodes' capacity to connect multiple sensing probes.	Hardware	[23]
Node Placement	A WSN topology is a very important factor for the network's dependability, fault tolerance, field coverage, reliability, etc. Moreover, physical aspects of the crops and environment must be considered since they will affect the performance of the network. Thereby, the nodes' placement should be carefully designed and smartly implemented.	Hardware	[23], [61]
Standard Framework	A standard methodology or framework for designing and implementing agricultural WSN could lead to more effective and interoperable systems. Such a framework should consider the different stages of precision agriculture, from the data gathering to the decision support.	Data	[23]
In-field Data Processing	Since data transmission is the most energy-consuming operation of the sensor nodes, the agricultural applications requiring high sensing rates must not deliver all the gathered data to preserve the WSN lifetime. Moreover, large amounts of raw data do not imply large amounts of useful information. Therefore, the nodes could process the gathered data to reduce the amount and frequency of the transmissions. This processing could include data aggregation to transmit only highly valuable information, and internal event detection to transmit only alerts in risky situations.	Data; Software	[6], [61]

In the second place, based on the research of Zeng *et al.* ([62]), Rahman *et al.* ([63]), Kim *et al.* ([64]), Jabeen and Nawaz ([6]), and Panigrahi *et al.* ([65]), we state the main issues challenging the execution of aggregation operations inside and among the nodes in WSN applications (Table 3.3). These issues are important for our conceptual-modelling process since data aggregation presents major possibilities, challenges and requirements when executed in distributed systems like WSN.

Table 3.3. Issues for executing aggregation operations in WSN

Issue	Description	Relation	Reference
Energy Consumption	This issue is challenging in agriculture and for aggregation in WSN. Since processing the data inside the sensor nodes consumes energy, the aggregation operations should reduce the transmitted data to save energy and increase the nodes lifetime.	Data; Software	[6], [62]–[65]
Routing	This issue is challenging in agriculture and for aggregation in WSN. If all the sensor nodes in a large WSN deliver their aggregates at the same time, some packages will not reach the base station. Therefore, the aggregation operations should consider some inter-node transmission protocols to avoid this issue.	Telecommunications	[6], [65]
Real Time	This issue is challenging in agriculture and for aggregation in WSN. The delivered data (aggregated or not) must be managed inside an opportunity window (*i.e.* real time); thus, the aggregation operations should increase the value of the delivered data and reduce its volume without affecting its opportunity. Besides, considering the routing challenge, the aggregation scheduling must avoid congestion while ensuring opportunity.	Data; Telecommunications	[6], [64], [65]
Synchronization	This issue is related to the routing and the real time, especially for inter-node aggregation. A timely inter-node aggregation requires that all the data reach the aggregator node inside the opportunity window; however, the nodes should not send their data at the same time. Therefore, a good synchronization is important to avoid congestion and allow the aggregator node to aggregate and deliver the data in real time.	Data; Telecommunications	[6]
Scalability	This issue is challenging in agriculture and for aggregation in WSN. Aggregating the sensed data inside the network is very important for its scalability since it reduces the volume of the managed data and increases the data value. Nevertheless, inter-node aggregation can become difficult in large WSN due to the complexity of synchronization.	Telecommunications	[6]

Propagation Losses	This issue is challenging in agriculture and for aggregation in WSN. A very important factor to consider the quality of the delivered (aggregated) data is the network status. A good Quality of Service in the WSN allows for well-transmitted data, which can be used in further inter-node aggregation operations and decision support.	Data; Telecommuni cations	[6]
Fault Tolerance	This issue is challenging in agriculture and for aggregation in WSN. The intra-node and inter-node aggregation reduce the transmission load of the network and the energy waste, reducing the probability of faults. Nevertheless, the aggregation processes should not affect other fault-tolerant mechanisms like node redundancy or topology control.	Data	[6], [62]
Reliability	This issue is challenging in agriculture and for aggregation in WSN. Aggregation is good for the reliability of the sensed data since it helps to reduce the noise that the sensors' hardware might introduce in the gathered measurements. However, the aggregated data must be dependable for the users since they will not be able to analyse the particular meta-data of each datum to check its quality. Therefore, the aggregation operations inside the nodes and among the nodes should consider the quality of each aggregated datum to provide the users with good-quality data.	Data; Hardware	[6], [62]
Usability	The aggregated information must be useful for the users, *i.e.* the aggregation process must consider the users' needs in the application context. Moreover, the user experience is important for the effectiveness of a decision support system.	Data	[6]
Heterogeneity	This issue is challenging in agriculture and for aggregation in WSN. Inter-node aggregation could leverage the network heterogeneity to storage and aggregate the data in the most powerful nodes. However, aggregating heterogeneous data is challenging, especially with different sensor nodes measuring the same variables with different configurations (*e.g.* gathering frequency). Furthermore, similar data from different contexts or environments should not be aggregated without special considerations, *e.g.* the internal temperature of the node and the air temperature are measures of the same kind that should not be combined.	Data	[6], [65]

Privacy	This issue has low relevance in common agricultural WSN. However, for more sensitive applications, inter-node aggregation must protect the private information of the node since pirate nodes might attempt to steal sensitive data.	Data; Software; Hardware; Telecommunications	[62]
Intruder Attacks	This issue has no major relevance in common agricultural WSN and is related to the privacy. Since pirate nodes could infiltrate the network to deliver erroneous data, inter-node aggregation should consider protection mechanisms against this kind of attacks.	Data; Software; Hardware; Telecommunications	[63]

Though all the challenges and requirements listed in Tables 3.2 and 3.3 are relevant, only those related to the WS data should be considered in this thesis; emphasising on the common issues.

3.3. List of Features

In this section, based on the analysis of the selected sensors and issues, we extract a list with some of the most relevant challenges, characteristics and constraints for the data modelling in agriculture-oriented WSN applications. These features could allow us to design the query processing in individual WS.

Moreover, aiming to present the WS data with a standard framework, we consider the IPSO guidelines for the definition of smart interconnected objects [66], specifically the sensors.

The extracted features of the nodes are:

- The measurement Type: indicates the type of sensed variable, *e.g.* temperature or humidity.
- The measured Value: indicates the obtained value of the measured variable, *e.g.* a temperature of 30.
- The measurement Units: complements the measured Value by indicating its units, *e.g.* the temperature Value could be in degrees Celsius or Fahrenheit.
- The measurement Location: indicates the location (coordinates) of the place where the node is located when the variable is sensed.

- The measurement Time: indicates the time (as a timestamp) when the variable is sensed.
- The LifeTime of the measurements: indicates how much time the measurements will remain stored in the node.
- The Granule for the measurements' LifeTime: complements the LifeTime of the measurements by indicating its units or time granule, *e.g.* seconds or minutes.
- The Maximum number of stored measurements: since the node will have limited resources, including storage, its maximum storage capacity should be considered to define the LifeTime.
- The measurement Quality: different data-quality indicators could be used to estimate the quality of the gathered measurements inside the node. This feature indicates such estimated quality of the data.
- The Optimal Range of the measure: indicates the range of the measured variable inside which it has high reliability. This could be a data-quality indicator.
- The Contextual Range of the measure: indicates the range of the measured variable outside which the values are extremely improbable. For instance, a temperature of 10°C in the South Pole is almost impossible nowadays; thus, such a reading cannot be trusted. This could be a data-quality indicator.
- The measuring Probe Position: indicates the relative position regarding the node of the probe that is sensing the variable, *e.g.* the height for air temperature or the depth for soil moisture.
- The measuring Probe Model: indicates the hardware model of the probe that is sensing the variable. This information could be important to check different data quality indicators of the specific probe.
- The measuring Probe Working Range: indicates the range of the measured variable outside which the probe measurements cannot be trusted. This could be a data-quality indicator.
- The measuring Probe Required Battery: indicates the minimum battery level that the probe requires to gather reliable measurements. This could be a data-quality indicator.
- The remaining Battery of the node: indicates the actual battery level of the node. Network managers must consider this feature to define preventive maintenance operations.
- The Battery measuring Units: complements the Battery level by indicating its units, *e.g.* voltage, amperage, or percentage.

- The Neighbours of the node: indicates the IDs of other WS that could communicate with the node considering the network topology, *e.g.* parent or child nodes.
- The Network Role of the node: indicates the role of the node in the WSN, *e.g.* a simple sensing node, an aggregator node, a cluster head, etc.
- The Position Configuration of the node: indicates whether the node is fixed on its position (*e.g.* a common rain sensor) or it is mobile (*e.g.* a sensor on cattle).
- The assigned Plot or crop of the node: indicates the ID of the plot or crop that the node is monitoring.
- The Communication Protocol: indicates the transmission protocol that the node is using to deliver the sensed data.
- The Communication Quality: indicates the QoS in the transmission of the sensed data. This feature could be based on different Link Quality indicators like the LQI or the RSSI.
- The node Operations: indicate the different operations that the node must run. For example, we could use three basic operations.
    - Data Gathering, the operation of sensing a variable from the environment.
    - Data Aggregation, the operation of aggregating the gathered information to reduce its volume and increase its value.
    - Data Delivering, the operation of transmitting the gathered or aggregated data to the base station or the next node.
- The Frequencies for Operating the data: each Operation is executed inside the node with some regularity; thus, a different frequency should be defined for each Operation.
- The duration Windows of the Operations: considering the sleep periods of the node, each Operation will work in different cycles. Therefore, the duration of each Operations' cycle can be defined as a Window.
- The Granule of the Operations' Frequencies and Windows: this feature complements the Frequency and Window of each Operation by indicating their units or time granule, *e.g.* minutes or hours.
- The Amount of data Operated in each Window: considering the sleep periods of the node, an Operation could be executed a limited number of times on each Window or cycle. This limit can be established by defining the maximum amount of measurements or data operated, beyond which the Operation enters in the sleep period until the Window finishes.

- The Aggregation Function: indicates the function that will be calculated from the gathered data in the Aggregation Operation, *e.g.* the maximum, minimum, average, etc.
- A separation between node-Internal and -External information: this separation feature should enable the implementation of privacy- and security-preserving politics inside the node. Moreover, it allows making a difference between the gathered data that exists only inside the node from the delivered information that the decision support systems can use.

Finally, after listing these features, we must analyse their relevance in order to select the most important characteristics and configurations for the design of the query processing and the temporal aggregation of the sensor measurements.

Summary

In this chapter, we have presented the process to discover a complete list of features for the description of the query processing and aggregation inside agriculture-oriented WS. These features are extracted from the analysis of the results of a literature review process, which allowed us to identify the most common agricultural sensor platforms, the most relevant issues for the implementation of WSN in agriculture, and the most important issues for executing aggregation operations inside WSN.

Although all the discovered features could offer a complete description of the WS data, conceptual models must be simple and concise representations [31]. Therefore, we must select the most relevant features from this list to build our conceptual model for the WS query processing; making it brief, comprehensive and useful.

Chapter 4

Conceptual Model

A conceptual model is a simpler and useful representation of a system under study, which can replace such system for some purposes, *e.g.* to assess the efficiency and effectiveness of a WSN before its implementation [31]. Thus, in this thesis, we focus on modelling the data collection and management features of the WS from the user point-of-view, which could allow for the design of the WS query processing.

Therefore, after the description and characterisation of the agriculture-oriented WS (Chapter 3), we must select the most relevant features regarding the design of the query processing and the temporal aggregation of the sensor measurements. Moreover, we must divide these relevant features into groups and define the relationships among them as an initial conceptual model. Finally, we must organise and categorise the selected features in each group, formalising the conceptual model in a standard modelling language (*i.e.* UML) [32].

4.1. Features Selection

Considering the importance of the modelling the data collection and management features of the WS for estimating their expected efficiency and effectiveness before the implementation, we have selected the most relevant characteristics and configurations for the design of the WS data from the user point-of-view.

Relevant data characteristics:

- The measurement Type.
- The measured Value (considering the Unit).
- The measurement Location.
- The measurement Time.
- The measurement's Estimated Quality.
- The remaining Battery (considering the Unit).
- The measurement Probe information (Position and Model).
- The Link Quality.
- The 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 the operation is 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 [7], [23], [58]. 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 Aggregated 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 features will allow us to model the data behaviour in agricultural WSN applications. However, they are not restricted to agriculture-oriented applications; thereby, these WS data features could be used to model WSN applications for various domains outside the Agri-food context.

4.2. Grouping and Representation

The selected features are explicitly classified in two main categories that underpin the grouping process: characteristics and configurations. The characteristics are features that describe the data in the WS, focusing on information useful to the system users (*i.e.* the queries). The configurations are features that describe the processing of the data in the WS, focusing on how each measurement is gathered, modified and delivered to the user (*i.e.* the queries processing).

Furthermore, we can divide the relevant characteristics in three ways:

- In the first place, the characteristics should be divided into two groups according to their purpose: the sensed data and the descriptive data. Since the main purpose of the WS is to transmit the sensed data, only the measured Value (considering the measurement Unit and Type) belongs to this group. The other characteristics can be considered as descriptive data that complement the sensed data, allowing for a better analysis.
- In the second place, the characteristics should be divided into two groups according to their relative importance for the models: the mandatory characteristics and the optional characteristics. Mandatory characteristics are related to the sensed data and need to be defined on any model. Only the measured Value (considering its Unit), the measurement Type, and the separation between Internal (Unavailable) and External (Available) data are mandatory. The other characteristics are optional since they complement the mandatory data, but the node could answer a query without configuring them.
- In the third place, the characteristics should be divided into three groups according to what they describe: the data, physical aspects, or meta-information. The Type, Value (with Unit), Location, Time and Estimated Quality describe aspects of the data. The Battery (with Unit), Probe Position and Model, and Link Quality describe physical aspects that might affect the data analysis. And the separation between Internal (Unavailable) and External (Available) data of the WS is meta-information.

However, we can also divide the physical and data characteristics into three groups according to the meta-information:

- Strictly internal data: this data only exists inside the node and is never delivered, though it might affect the delivered data. Only the measurement Probe information is strictly internal.
- Strictly external data: this data does not exist inside the node for a long time and is only important after it is delivered. Only the Link Quality is strictly external.
- General (internal and external) data: this data is important inside and outside the node; however, it could be defined as strictly internal or external in specific models. The Type, Value (with Unit), Location, Time, Battery (with Unit) and Estimated Quality are general characteristics.

Similarly, we can divide the relevant configuration in two ways:

- In the first place, the configurations should be divided into four groups according to their purpose: configure the measures, the gathering operation, the aggregation operation or the delivering operation. The measure configurations describe the data regarding its processing in the node, only the LifeTime (considering its Granule) belongs to this group. The gathering configurations describe how the data is gathered (sensed) from the real world; the Gathering Frequency, Window, Granule and Amount belong to this group. The aggregation configurations describe how the data is aggregated in the node and delivered to the user; the Aggregation Function, and the Aggregating Frequency, Window, Granule and Amount belong to this group. The delivering configurations describe how the data is delivered to the user; the Delivering Frequency, Window, Granule and Amount belong to this group.
- In the second place, the configuration should be divided into two groups according to their relative importance for the models: the mandatory configurations and the optional configurations. The only mandatory configuration is the Aggregation Function for an aggregation operation. Nevertheless, since the node aim is to transmit the sensed data, a node must define a gathering and a delivering (or aggregation) operation for each measure.

All these groups of characteristics and configurations are very important for the conceptual model. However, since the gathering operation is more related to the internal data, and the aggregation and delivering operations are more related to the external data, an initial representation considering these groups is easier to understand (Figure 4.1).

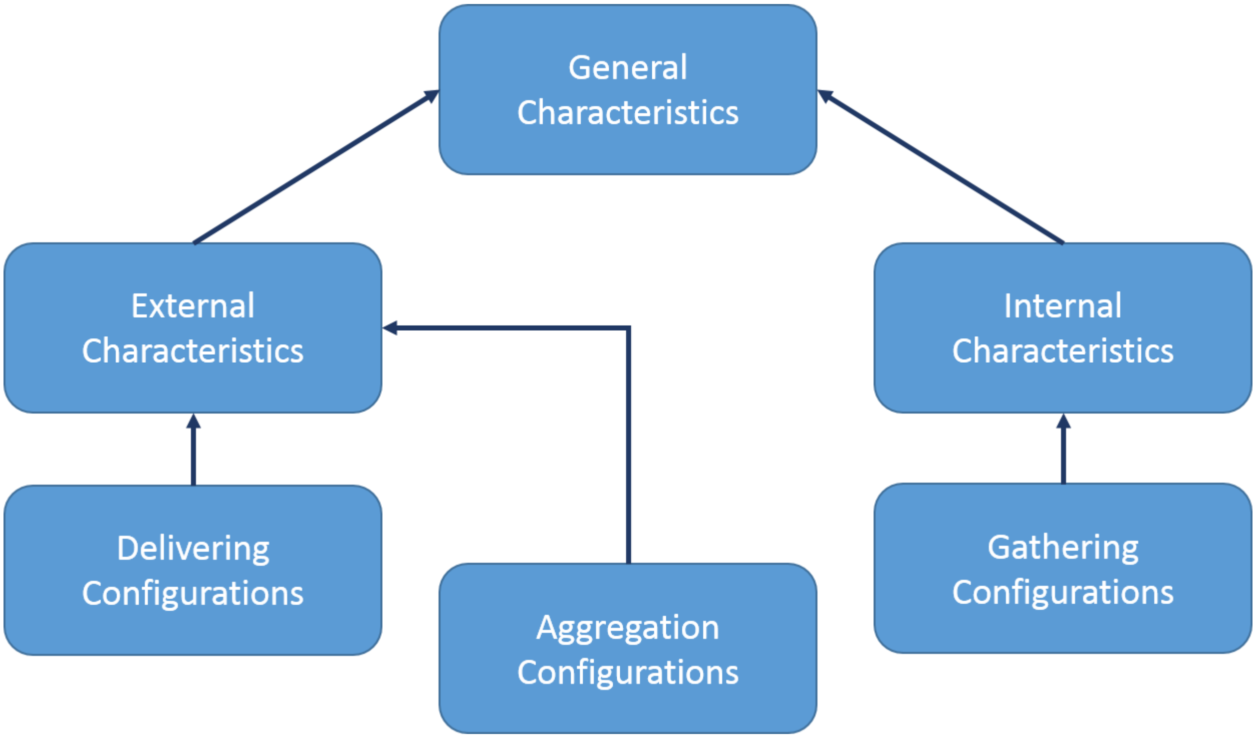

Figure 4.1. Initial representation of the grouped features and their relationships.

4.3. UML Meta-Model

Based on the most relevant WS features, their classification, and the initial representation of the groups (Figure 4.1), we define a meta-model in UML. This meta-model allows for the design of the query processing in individual Wireless Sensor Nodes in agricultural applications, especially considering the user point-of-view. Moreover, we complement our meta-model with some profile usage examples.

4.3.1. Data-Centric Wireless-Sensor UML profile

In the UML context, meta-model design is achieved through considered extensions. These extensions can be heavyweight (when the semantics are changed by changing the standard UML meta-model), or lightweight (when the semantics are adapted without changing the standard UML meta-model) [67]. Since lightweight extensions preserve the UML standard they are easier to learn and understand; thus, we design our meta-model as a lightweight extension, *i.e.* a UML profile.

The purpose of UML profiles is to allow customising UML for particular domains or platforms by extending its meta-classes (class, property, etc.) [68]. A profile is defined using three key concepts: stereotypes, tagged values and constraints. A stereotype extends a UML meta-class and can be represented using the notation «stereotype-name» 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 in 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".

In our Data-centric Wireless-Sensor UML profile (Figure 4.2) we define 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 three types of constraints. This profile (Figure 4.2) is a framework for modelling the data behaviour in WS implemented on Agri-food-oriented ICT applications.

The data types in our profile (Figure 4.2) 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 Vivas *et al.* ([69]), 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.

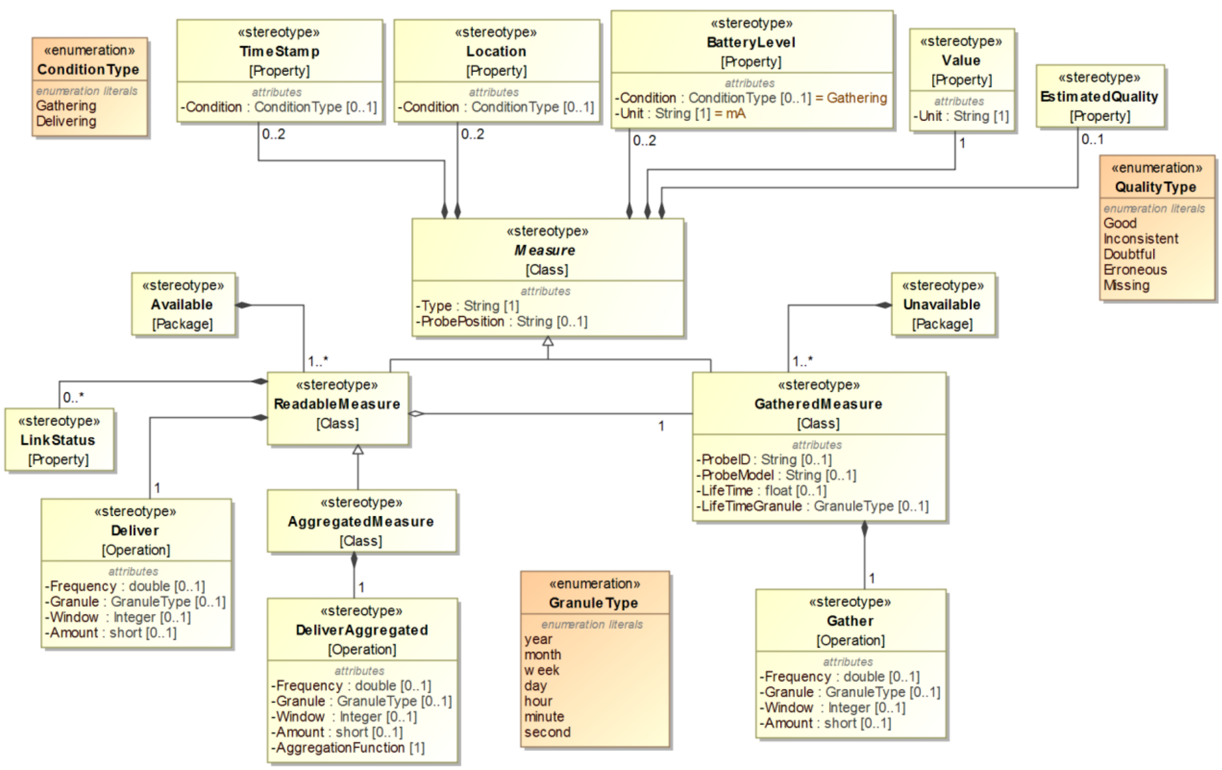

Figure 4.2. Data-centric Wireless-Sensor UML profile from the user point-of-view.

The main root of our meta-model 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 with 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.
- 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**, leveraging proposals like [69].

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.

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.

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 of 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 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.

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 [70]. 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 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.

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

- *Meta-model level constraints*: these constraints are defined at the meta-model level and grant well-formed class diagrams using our UML profile. Example 6 presents two OCL rules of this type.
- *Semantic coherence constraints*: these constraints are associated with 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 to or greater than the Gathering period (1/FG);
  - the Window (Win) on each operation must be equal to or greater than the operation period (1/F);
  - the total amount of stored measurements (ΣSM) 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 (*e.g.* to deliver only good quality data). Example 8 presents some OCL rules of this type for the hypothetical case studies of Examples 3 and 4.

This way, our profile can be used to model different agricultural applications regarding the WS and WSN data behaviour. These application models would allow evaluating its efficiency and effectiveness before its development and implementation. Furthermore, these models could be leveraged to generate code for programming the nodes [31]. Hence, in Appendix A, we present an initial, untested algorithm for transforming our profile models into JavaScript Object Notation (JSON) configuration files.

4.3.2. Profile use examples

We propose eight examples of the use of our profile (Figure 4.2), which include the modelling in UML of the measures, simple nodes and aggregating nodes, and the modelling in OCL of the constraints.

Example 1 - GatheredMeasure

The Class SoilMoisture0 (Figure 4.3) 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 describing 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).

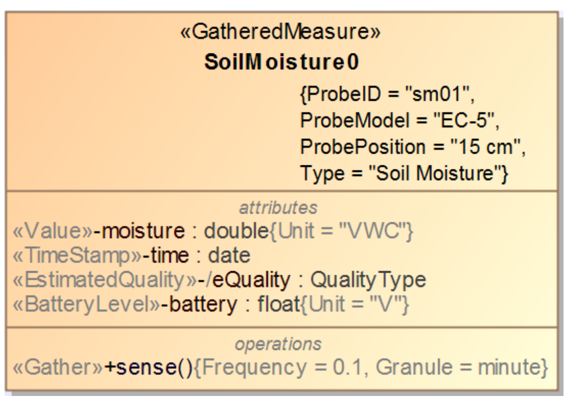

Figure 4.3. Example Class implementing the **GatheredMeasure** stereotype

Table 4.1 contains an example of the data represented by SoilMoisture0 (Figure 4.3). This data model allows the sensor node to gather one Moisture measurement (recording Time, Quality and Battery) every 10 minutes.

Table 4.1. 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

Example 2 - RedableMeasure

The Class 3SoilMoisture (Figure 4.4) 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 every 10 minutes), but only a maximum amount of three values are delivered inside each 60-minutes window.

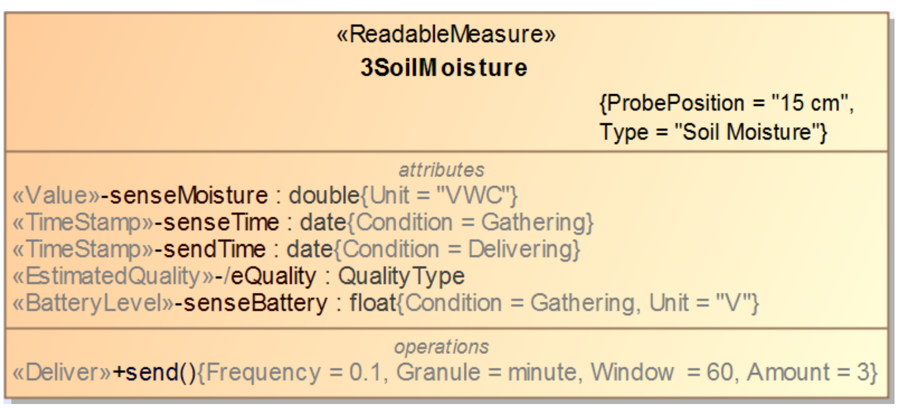

Figure 4.4. Example Class implementing the **ReadableMeasure** stereotype

Table 4.2 contains an example of the data represented by 3SoilMoisture (Figure 4.4). This data model allows the node to deliver one Moisture measurement (including Times, Quality and Battery) every 10 minutes, with a maximum of three measurements per hour. For example, data is delivered during the 22nd hour at 22:03; 22:13 and 22:23.

Table 4.2. Example data for 3SoilMoisture

Sense Moisture	Sense Time	Send Time	EQuality	Sense Battery
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 - Simple node

These classes (Figure 4.3 and Figure 4.4) could represent a single-node application example (Figure 4.5), on which the hypothetical user (*e.g.* a farmer) needs to know the soil moisture 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.

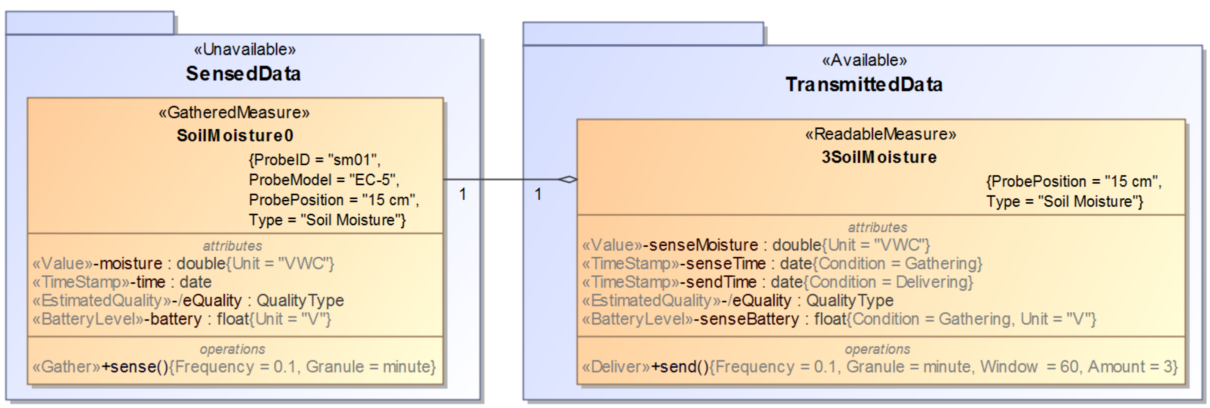

Figure 4.5. UML model of a moisture WS data from the user point-of-view.

Tables 4.3 (Gathered) and 4.4 (Delivered) contain an example of the data represented by this model (Figure 4.5). These data show that the node gathers one Moisture measurement (recording Time, Quality and Battery) every 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 16th hour (yellow lines of Table 4.3), only three values are sent (yellow lines of Table 4.4).

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.

Table 4.3. 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

Table 4.4. 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

Example 4 - AggregatedMeasure

The Class AggregatedSoilMoisture (Figure 4.6) 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 4.6) represent data accessible for the application or the final user. These attributes are related to the **GatheredMeasure**. However, unlike in the 3SoilMoisture example (Figure 4.4), 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.

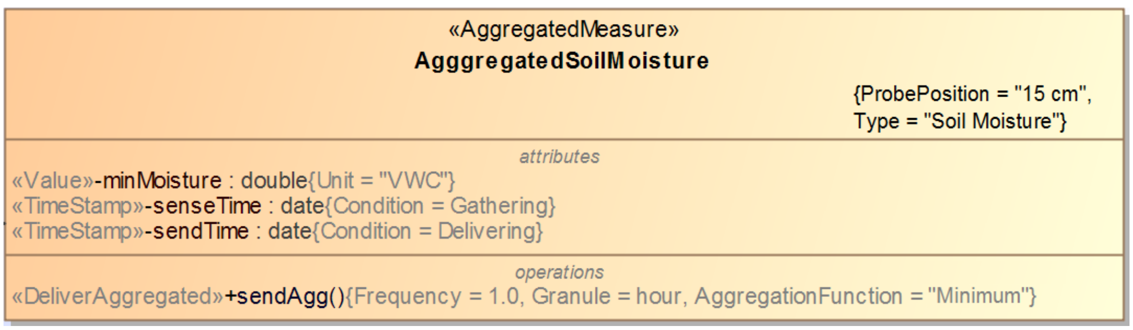

Figure 4.6. Example Class implementing the **AggregatedMeasure** stereotype

Table 4.5 contains an example of the data represented by AggregatedSoilMoisture (Figure 4.6). 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.

Table 4.5. 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

Example 5 - Aggregating node

Another application example (Figure 4.7) could leverage the **AggregatedMeasure** 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 4.7) is similar to the first one (Figure 4.5) with one important difference: the user only requires aggregated data.

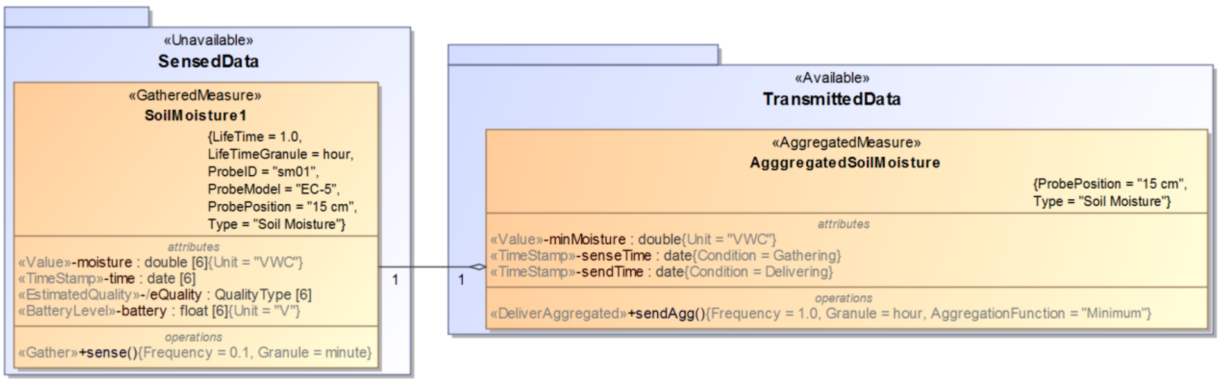

Figure 4.7. UML model of a moisture WS data with aggregation from the user point-of-view.

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 4.6 (Gathered) and 4.7 (Delivered) contain an example of the data represented by this model (Figure 4.7). These data show that the sensor node gathers one Moisture measurement (recording Time, Quality and Battery) every 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 7th hour (yellow lines of Table 4.6) only the one with the minimum value (40) and Good quality is sent (yellow line of Table 4.7).

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).

Table 4.6. 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
…	…	…	…

Table 4.7. 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
…	…	…

<u>Example 6 - Meta-model level constraints</u>

In this example, we present two meta-model level OCL constraints. In the first place, 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 4.8).

Context	Measure
Name	Measure-Value attribute
OCL	self.ownedAttribute ->select (m | m.oclIsTypeOf(Value))->size()=1
ErrorMessage	Every <<Measure>> must contains one, and only one <<Value>> attribute.

Figure 4.8. OCL for meta-model level constraints regarding the obligatoriness of a <<Value>> attribute in all the <<Measure>> classes.

In the second place, the rule specifying that any class stereotyped as <<GatheredMeasure>> must have one (and only one) operation stereotyped as <<Gather>> (Figure 4.9).

Context	GatheredMeasure
Name	GatheredMeasure operation
OCL	self.ownedOperation ->select (g | g.oclIsTypeOf(Gather))->size()=1
ErrorMessage	Every <<GatheredMeasure>> must contains one, and only one <<Gather>> operation.

Figure 4.9. OCL for meta-model level constraints regarding the obligatoriness of a <<Gather>> operation in any <<GatheredMeasure>> class.

Indeed, each one of the three class stereotypes: ReadableMeasure, GatheredMeasure and AggregatedMeasure, must have one (and only one) specific operation: Gather, Deliver and DeliverAggregated, in the same order. Hence, we define similar rules for each stereotype in the profile.

Example 7 - Semantic coherence constraint

In this example, we present the OCL for some semantic coherence constraints; in particular for the lifetime granularity (Figure 4.10).

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

Figure 4.10. OCL for semantic-coherence constraints regarding the granularity of lifetime in GatheredMeasure.

This constraint (Figure 4.10) indicates that designers should define both the LifeTime and LifeTimeGranule tags in the GatheredMeasure class if they want to have persistence in the gathered data.

Example 8 - Application-specific constraints

In this example, we present some user-defined constraints. Considering the aforementioned application examples (Figure 4.5 and Figure 4.7), designers will need to define application-specific constraints in OCL for each case. The first application (Figure 4.5) 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 4.11).

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 4.11. OCL application-specific constraints for example 3.

The first constraint in Figure 4.11 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 4.7) requires to deliver only good-quality data. Thus, it needs to identify the quality of the data and include only good-quality values for aggregation (Figure 4.12).

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

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.

Figure 4.12. OCL application-specific constraints for example 5.

The first constraint in Figure 4.12 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 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.

Summary

In this chapter, we have presented and explained our conceptual model for the design of the query processing in WS. In the first place, from the results of Chapter 3, we have selected the most relevant features for describing the data in WS, considering the user point-of-view. In the second place, we have classified the 24 features according to their focus, purpose and relative importance. Furthermore, we have grouped, organised and related the features in a basic conceptual model (Figure 4.1). Finally, we have formalised the conceptual model as a meta-model in UML (profile), which allows for a clear description of the data processed by WS from the user point-of-view (Figure 4.2). Besides, aiming to provide a better understanding of our UML profile, we have presented and explained different examples of use.

The design of the WS data behaviour 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. Furthermore, these models could be leveraged in Model-Driven approaches to create or automate the execution of software-based systems through complex techniques like meta-modelling, model transformation, code generation or model interpretation [31].

Chapter 5

Validation

According to Rodrigues da Silva ([31]), a model must allow for the identification of the original system, must be a simplified version of the original, and should be able to replace the original for certain purposes (*e.g.* analysis). Moreover, since models are described in a modelling language, they must also comply with the language-specific structure and constraints [10], [31].

Therefore, in this chapter, we validate our UML profile with three different strategies: Academic Validation, which allows us to assess if the conceptual model makes sense for academics and practitioners involved in the design and implementation of WSN and other kinds of information systems [32], [71]. Case Study, which allows us to estimate the feasibility and pragmatism of the conceptual model [31], [32], [71]. And CASE-Tool Validation, which allows evaluating the correctness and consistency of the profile and its models in the UML standard [10].

5.1. Academic Validation

This validation strategy consists of presenting a theoretical model to different experts, scholars and practitioners in an academic framework (*e.g.* a seminar) and receiving their comments and feedback. It aims to assess if the proposed model presents a reasonable theory for different academics working on similar issues [32]. This strategy allows us to evaluate if the model enables the identification of the original system (*i.e.* the mapping criteria) and provides an initial insight of the model's usefulness (*i.e.* the pragmatism criteria) [31].

Although this kind of validation strategy is good to assess the opinion of the academy on the proposed model, it is highly subjective and could drive biased results [71], [72]. Hence, we have selected two different academic frameworks with different scholars and practitioners to increase the quality and reduce the risk of bias in this validation: Expert panel and Seminar.

5.1.1. Expert panel

Based on the guidelines provided by Kitchenham *et al.* ([71]) for the collated expert opinion, we presented our data-centric UML profile for agricultural WS to three experts on ICT for agriculture from the Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture (Irstea) and the Université Clermont Auvergne (UCA):

- Dr Gil de Sousa[1], Research Engineer, COPAIN team, Irstea, France.
  Expert in the design and implementation of WS and WSN for agriculture.
- Dr François Pinet[2], Research Director, COPAIN team, Irstea, France.
  Expert in conceptual modelling applied to agricultural and environmental systems.
- Prof Michel Schneider[3], Professor Emeritus, UCA, France.
  Expert in data modelling and information systems.

Firstly, the experts received a brief introduction to the context of this thesis, especially focusing on the importance of modelling the data querying in agricultural WS. Secondly, they received a complete explanation of the developed UML profile, including the WS

[1] DBLP profile: http://dblp.org/pers/hd/s/Sousa:Gil_De
[2] DBLP profile: http://dblp.org/pers/hd/p/Pinet:Fran=ccedil=ois
[3] DBLP profile: http://dblp.org/pers/hd/s/Schneider:Michel

feature description and selection, and some application examples. Thirdly, each one of the experts gave us their opinion about our profile from the field of their expertise. Finally, we discussed with the experts about the strong and weak points of the model, its usefulness, its limitations, important improvements, and possible future research. This last step was developed as a dialogue in the form of a focused observation [72], recording all the relevant comments and suggestions in a log.

Overall, the experts agreed on the importance of modelling the WSN data for any kind of application, not only the agriculture, and that our model constitutes a very important first step towards the automatic programming of WSN and the conceptual integration between WSN and other kinds of information systems (*e.g.* OLAP or DBMS). They also agreed that our model is clear enough to be used on a model-driven approach for WSN.

Nevertheless, the experts also expressed us all their doubts, advice, and discrepancies (Table 5.1).

Table 5.1. Experts' suggestions on the UML profile

Expert	**Doubts, advice, and discrepancies**
Dr Gil de Sousa	1. There should be general tags like the measure type. 2. Measuring probes might also have some spatial position (*e.g.* Height). 3. Sometimes, knowing the probes' model can help to define the quality assessment. 4. Maybe, battery and location could be considered measures. 5. A model for text implementation could have high relevance.
Dr François Pinet	1. The model does not allow for spatial aggregation. 2. The model can be too rigid for modifications on-the-march. 3. This work should be submitted to a peer-reviewed journal for wider academic validation. 4. There are no relationships among the measures.
Prof Michel Schneider	1. The "sending measure" could have small relevance on our model. 2. The aggregates could be calculated and stored inside the node. 3. The aggregated data could be re-aggregated (distributive and algebraic, not holistic). 4. There are no relationships among the measures.

Considering the scope and objectives of this thesis, and with the approval of the experts, we decided to classify their suggestions into three categories: imperative, relevant, and unconsidered.

Imperative suggestions

These suggestions are in the scope of this thesis and are very important for the achievement of its objectives. Thus, we must review our profile according to these suggestions before considering it as “validated by experts”.

The changes derived from the imperative suggestions are:

- Definition of a relative-position tag: according to Dr de Sousa, the measurements are gathered from a spatial position relative to the location of the node, *e.g.* a height above the node or a depth into the ground. Therefore, we define a ProbePosition tag, which enables the configuration of such relative position.
  This tag is very useful when one node gathers multiple measurements of the same type from different relative positions. For example, the soil moisture is usually gathered from different depths to estimate the water that each part of the plant roots will receive; thus, considering the depth of each measurement is important for the data analysis.
- Definition of general tags: according to Dr de Sousa, some meta-information (illustrated in the model as tagged values) should be defined for both the node-internal (unavailable) data and the node-external (available) data. Therefore, we define two tags in the Measure abstract class: Type, which defines the measurements type (*e.g.* temperature or humidity). And ProbePosition, which defines the relative position of the measurements.
- Definition of a probe identification tag: according to Dr de Sousa, the quality of the measurements also depends on the measuring hardware (*i.e.* the probe), and one way of identifying how the hardware affects the data is through its model. Therefore, we define the hardware model meta-information in the GatheredMeasure class as a ProbeModel tag.

With the implementation of these changes in the model, our data-centric UML profile for agricultural WS can be considered as complete for the scope and objectives of this thesis. Besides, it can also be considered as “validated by experts”.

Furthermore, we also considered as imperative the advice of Dr Pinet regarding the submission of this thesis’ results to a peer-reviewed journal. Hence, we have submitted a paper relating the main outcomes of this thesis to the International Journal of

Agricultural and Environmental Information Systems (IJAEIS), which is indexed in the Web of Science ® and Scopus ®. The submitted paper is found in Appendix B.

Relevant suggestions

These suggestions are out of the scope and objectives of this thesis. However, they should be considered to define future related research or as important extensions for this thesis.

The suggestions that we should consider for driving new research in the short and medium term are:

- The definition of a model-to-text mapping mechanism (recommended by Dr de Sousa). Automatic code generation for programming WS will be very useful for scientists and engineers working on WSN since this feature could ease many tasks in model-driven approaches. However, the automatic generation of code is a different problem than the modelling of sensor queries; hence, we cannot consider it directly in this thesis.
  An initial (untested) prototype of a mapping algorithm for our model is found in Appendix A.
- The definition of relationships between measures (recommended by Dr Pinet and Prof Schneider). Allowing the measures to relate between them could give our profile a high dynamism, enabling the modelling of more diverse and complex systems. Though the definition of such a relationship is apparently as simple as drawing a line in the CASE tool, the implications of such join operation will require a deeper study on the subject and surpass our analysis objectives. Hence, we consider this suggestion is not convenient in the scope of this thesis; nevertheless, we should consider it in future research projects.
- The definition of a spatial aggregation mechanism (recommended by Dr Pinet). Considering that one node can gather multiple measurements of the same type from different relative positions, the aggregation of the gathered data in the spatial dimension besides the temporal dimension is very interesting. However, aggregating data in multiple dimensions is a complex task that might overwhelm the processing capabilities of some sensor platforms. Moreover, the (spatial) aggregation of different measures requires the definition of join operations, which are beyond the scope and objectives of this thesis.

- The definition of a mechanism for re-aggregating distributive and algebraic aggregates (recommended by Prof Schneider). One of the advantages of distributive and algebraic aggregation operations (*e.g.* sum, count, average, or standard deviation) is that they can compute the data in separate sets without affecting the final aggregate. For example, a system calculating the total production of one day could store each production datum of the day and sum all the values at the end of the day (high memory requirement); sum the production data each hour, store only the calculated values, and sum those values at the end of the day (medium memory requirement); or simply sum each production datum as it arrives, delivering the total sum at the end of the day (low memory requirement). This re-aggregation process could be especially useful for WS, where the memory and computing resources are limited. However, the implementation of such mechanism in the model requires a clear separation between the aggregation types, the definition of specific rules and constraints for each aggregation operation, the definition of a node-storage mechanism, and the definition of join operations. These changes would highly increase the complexity of our profile, their implications require a deeper study on the subject, and their implementation is not strictly inside the scope and objectives of this thesis. Hence, since these changes are not necessary and such complexity increase is not convenient, we should only consider this suggestion in future research projects.
- The definition of a mechanism for calculating and storing the aggregates in the node (recommended by Prof Schneider). Though this is an interesting suggestion, its implementation requires the definition of join operations, which surpass the scope and objectives of this thesis. Moreover, we consider that this mechanism is a prerequisite for the definition of a re-aggregation process for distributive and algebraic operations. Since we do not address the problem of re-aggregation in this thesis, such storing mechanism will have little to no relevance. Therefore, this suggestion should only be considered when addressing the problem of the re-aggregation process for distributive and algebraic operations.
- The consideration of some properties (*e.g.* the battery or the location) as individual measures (recommended by Dr de Sousa). Defining these properties as measures could increase the dynamism of our model. Indeed, from a physical point of view, these properties are also measured by the nodes with different

equipment like GPS. Notwithstanding, considering these properties as individual measures brings two problems:

- ○ The complexity of the profile will increase, not only for the definition of different types of measures, but because these measures should be related with join operations, which definition is beyond this thesis.
- ○ The objective of the profile could be lost. Although the battery and the location of a WS are important measures that allow for a better description of the data (*e.g.* georeference, quality assessing), the objective of the sensors is to measure the real world. For example, a hardware platform capable of acquiring its GPS position and its remaining battery, but incapable of measuring other variables (*e.g.* temperature or humidity) is not considered as a sensor platform, except in very specific applications. Thus, such auxiliary measures must not be considered on the same level than the main measures.

We must carefully analyse and solve these issues before considering this suggestion for the development of new research.

Though being relevant to the profile, these suggestions are beyond the scope of this master's thesis and must not be considered as prerequisites for the validation of our profile. Nevertheless, we should consider them for the development of new research projects around this topic.

Unconsidered suggestions

These suggestions are out of the scope and objectives of this thesis. They were generated by some confusion of the experts, lack of information, or misunderstanding of the model or the speech. Thus, we clarified them in the dialogue with the experts, who agreed on the confusion.

These suggestions do not affect the validation of our profile. However, they must be considered to improve the way we present our profile in the visual and auditory aspects. These considerations are:

- The apparent irrelevance of the “sending measure”. Due to the separation of the node-internal (unavailable) and the node-external (available) information the measures must also be separated. However, the name of the available measures “sending measure” was not descriptive, and the data separation had no relevance in our speech. Therefore, we changed the stereotype name to “ReadableMeasure”, and added more importance to the data separation in our speech.

- The low flexibility of the profile models. The profile models are very rigid; however, this is good. The WS are programmed in a laboratory before being deployed in the fields, and they are rarely re-programmed after the establishment of the WSN. Indeed, most agricultural WSN are intended to last long periods of time without requiring any attendance from the technicians or the engineers, and their program is almost never modified. Thus, a rigid model is adequate for rigid systems, although we had overlooked this feature in our speech.

From the results of this expert panel (Table 5.1), their analysis and discussion, and our interpretation of this validation process, we conclude that: after the imperative changes, our profile can provide a better and clearer description of the query processing inside agricultural sensor nodes. The experts could identify the original system (*i.e.* the sensor platform) through our conceptual model, which validates its mapping criteria [31]. The experts found our model useful, which helps to validate the pragmatism criteria [31]; indeed, all the relevant suggestions reflect the potential of the profile.

5.1.2. Seminar

Following the validation guidelines of Jabareen ([32]) for conceptual representations, we presented our data-centric UML profile for agricultural WS in the Research Seminar space of the *Maestría y Doctorado en Ingeniería Telemática* at Universidad del Cauca, Colombia.

This seminar is an academic framework for all the students of the Master’s and Doctoral programs in Telematics Engineering. Also, some teachers and engineering students usually attend the seminar. Most of the students work on applied science and engineering, and many of them are currently working or expecting to work with sensor platforms and WSN in different domains, *e.g.* health, agriculture, tourism, environment, etc*.* Thereby, we consider this seminar as a good academic framework to assess the opinion of different scholars and practitioners on our model; complementing the results of the expert panel, and increasing the quality and reducing the risk of bias in the validation of our profile [71], [72].

Firstly, we gave the seminar attendees a brief introduction to the context of this thesis, especially focusing on the importance of modelling the data querying in agricultural WS, our research scope and objectives, and the current state of the art. Secondly, we thoroughly explained our UML profile, including the WS feature description and selection, its development, and some application examples. Thirdly, we presented the conclusions and proposed future works obtained from the validation process with the experts in France. Fourthly, the attendees freely expressed us their opinion, doubts, and discrepancies about our profile, considering their own research goals and point of view. Finally, we joined into a discussion with the most interested attendees about the strong and weak points of the model, its usefulness, its limitations, important improvements, and possible future research. This last step was developed as a dialogue in the form of a focused observation [72], recording all the relevant comments and suggestions in a log.

In general, the master's and doctoral students expressed that the design of the query processing in WS is an interesting topic and our profile seems to be a good option for such purpose. They asked about the definition of new models from the profile for different applications, and their validity outside the agricultural domain. Their main discrepancy was about the size of the screen since it was not big enough to appreciate all the profile in a single image.

In the discussion with the most interested attendees, they highlighted four advantages of our profile:

- It describes the data inside and outside the sensor platforms, which helps to clarify the processing of the queries.
- It is a very good option for formalising the design and implementation of WS applications; it can be used for writing manuals, papers, reports, etc.
- With this kind of models, using model-driven approaches for the definition and implementation of WSN becomes easier.

- In development projects, this profile enables the definition of WSN models that can be discussed with other team members, the advisors, and the clients before the actual implementation.

Finally, some of the attendees expressed their intention to use our conceptual model in their research projects.

Therefore, based on these results, we conclude that the seminar attendees could identify the original system (*i.e.* the sensor platform) through our conceptual model, validating its mapping criteria [31]. Also, they found our model useful, which provides a good hint in the validation of its pragmatism criteria [31]; indeed, they proposed different situations close to their contexts whereby our profile could help them.

Furthermore, regarding the results from the expert panel and the seminar, we have validated our data-centric UML profile for agricultural WS (Figure 4.2) in two different academic frameworks, with different scholars and practitioners from different contexts. Hence, our conceptual model can be considered as “validated by the academy” in a reliable qualitative process [72].

5.2. Case Studies

After our profile proved that it is interesting and represents an important part of WS through a successful academic validation, we must prove that it can design the query processing in WS applications, validating its pragmatism criteria [31]. Hence, we evaluate the usefulness of our data-centric UML profile for agricultural WS in three real case studies [10], [31], [71]: one from Irstea and two from Universidad del Cauca.

5.2.1. Irstea - iLive network

The iLive network [73] is a result of a partnership between the Irstea institute and the LIMOS (*Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes*) laboratory. The goal 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. 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 constitutes 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 [74].

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 probes and four Irrometer Watermark probes; though not all these probes are connected to the nodes. For example, in this experimentation, nodes are only equipped with three Irrometer 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 5.1).

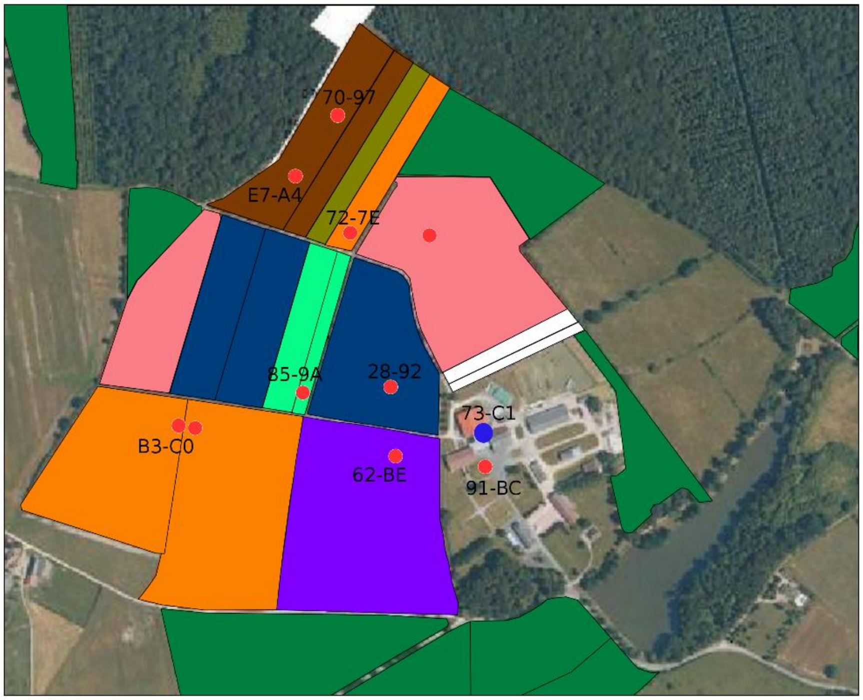

Figure 5.1. Deployment of the iLive network in the Montoldre site.

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, we analyse a small data subset delivered by one of the iLive nodes: the 91-BC (Table 5.2).

Based on the analysis of this data (Table 5.2), 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 5.2).

Table 5.2. Data subset for the analysis of the iLive network from the node 91-BC.

Node ID	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**

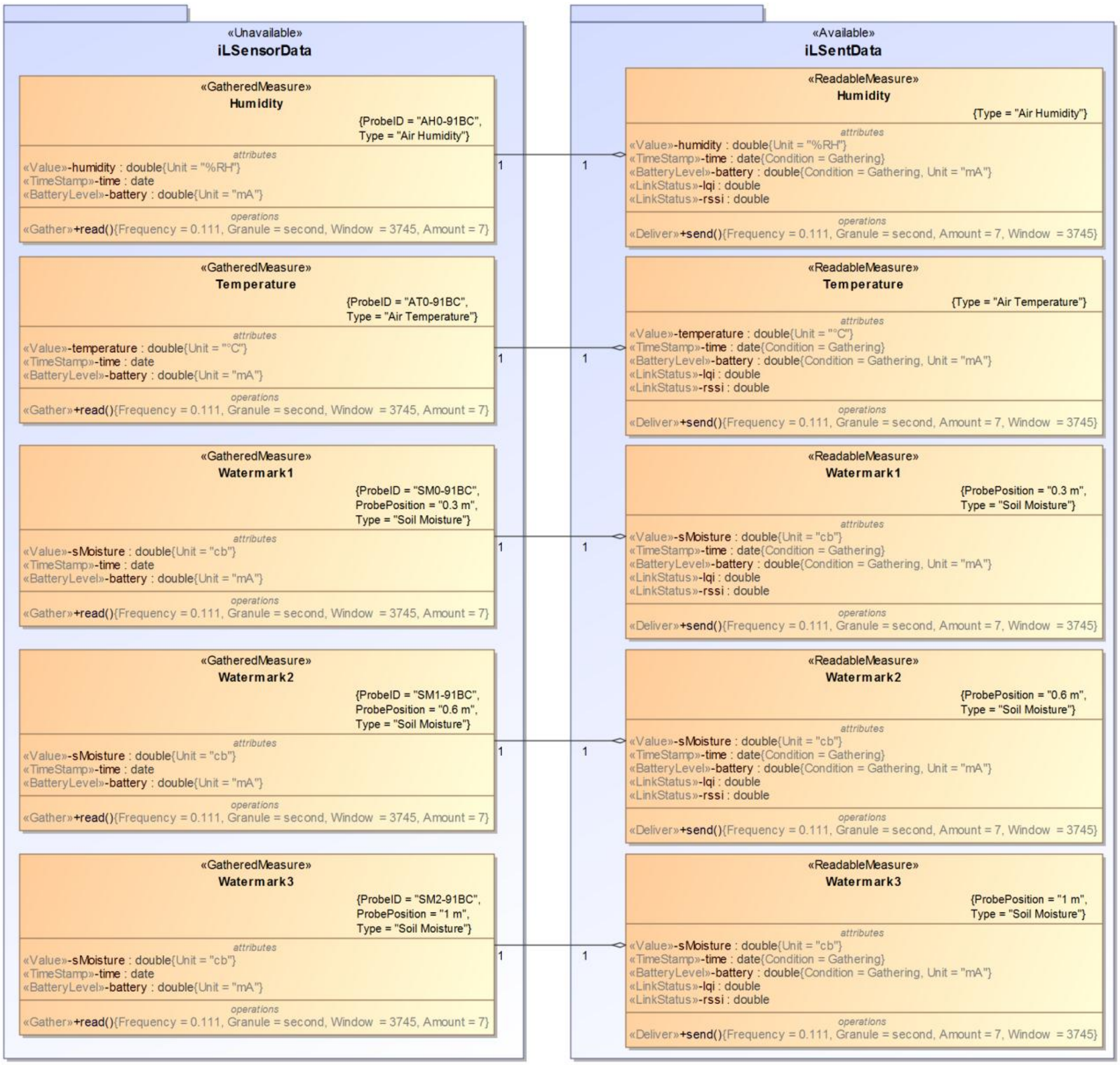

Figure 5.2. UML data model from the user point-of-view for the iLive case study, node 91-BC.

The modelled node (Figure 5.2) 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).

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 characterising 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 every 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 5.2) allows visualising 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, the amount of delivered measurements could have been reduced with aggregation functions like average, which helps to reduce the sensor noise and battery waste in the data transmission [7], [58], 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.

5.2.2. Universidad del Cauca - weather station

The weather station of the Universidad del Cauca is used by the Aquarisc project [75] for monitoring the meteorological conditions in the city of Popayán, especially rainfall. These rain data allow Aquarisc to evaluate their rain-amount forecasting systems for the High Colombian Andes, which would enable estimating the water level of some of the major rivers in Colombia [76].

The monitoring hardware is a DAVIS Precision Weather Station - Vantage Pro2 Wireless, which can measure different meteorological parameters like temperature, relative humidity, wind speed, dew point, precipitation, barometric pressure, etc., and also calculates multi-measure indexes like the heat index or the wind chill. In this case study, the station gathers, calculates and delivers all the measures and indexes every hour.

However, according to Valencia-Payan and Corrales ([76]), the most relevant data from this station are the temperature, precipitation (rainfall), and rain rate, since these measures are better for comparing their forecasting models.

Thereby, for modelling and validation purposes, and considering that our profile cannot model multi-measure calculations since these operations are not common in regular WS, we analyse a small data subset from the weather station considering only the most relevant measures (Table 5.3).

Table 5.3. Data subset for the analysis of the weather station

Date	Time	Temperature	Precipitation	Rain Rate
14/06/2016	05:00 a. m.	13.2	0.00	0.0
14/06/2016	06:00 a. m.	13.8	0.00	0.0
14/06/2016	07:00 a. m.	14.7	0.00	0.0
14/06/2016	08:00 a. m.	17.6	0.00	0.0
14/06/2016	09:00 a. m.	18.6	0.25	0.0

Based on the analysis of this data (Table 5.3), the mentioned station characteristics and use, and considering our profile, we propose the following UML model for the (simplified) description of the data in the weather station for rainfall measurement of the Aquarisc project (Figure 5.3).

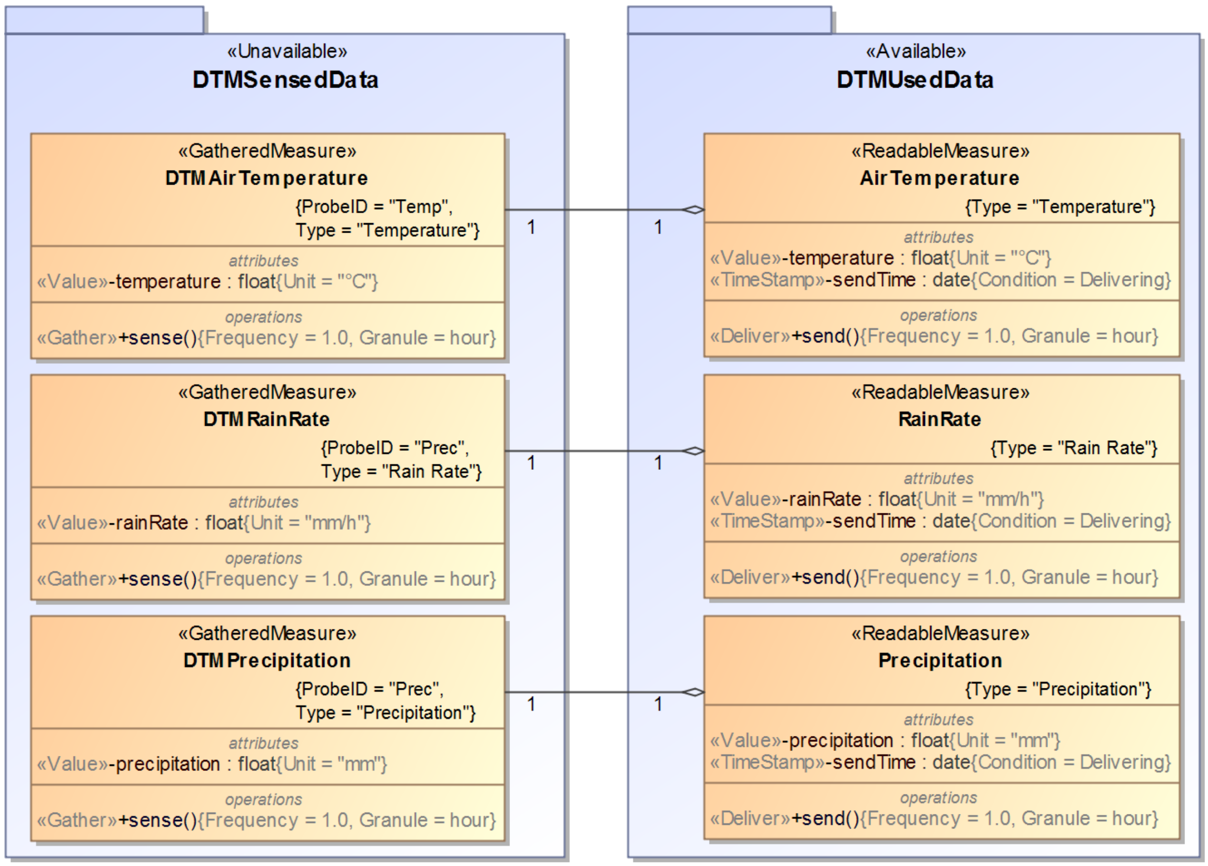

Figure 5.3. Simplified UML model of the Aquarisc station data from the user point-of-view.

The modelled station (Figure 5.3) gathers and delivers three types of measurements: Temperature in degrees Celsius (°C), Rain Rate in millimetres per hour (mm/h), and Precipitation in millimetres (mm). For these measures, only the measurement values are being gathered, *i.e.* no other information like battery or time is relevant for the node-internal data. However, the delivering timestamp is considered in all the measures for the node-external data.

Also, the station model (Figure 5.3) shows that all the measures are gathered and delivered with the same frequency: 1 measurement per hour, without a maximum number of measurements or a working window.

This visual model (Figure 5.3) could help Aquarisc members to formalise their reports and results regarding the use of the station data. Furthermore, it could also allow them to improve or even automatize their experiments' processes.

In this sense, Aquarisc members require processing the station data before they can use it with their models. This data processing consists of two main steps: in the first place, they find and eliminate outliers based on logical and contextual parameters for the Rain Rate and the Temperature. In the second place, they aggregate the hourly data of each day into daily values of average, maximum and minimum Temperature, total Rainfall, and average Rain Rate [76].

Therefore, Aquarisc members could leverage our UML profile to re-model the weather station, including the data aggregation and quality estimation processes. This data design would reduce the manual processes in Aquarisc and the computing load of the central storage server, improving the performance of their system and experiments.

Hence, Figure 5.4 is an example representation of the weather station data, according to the specific needs of the Aquarisc project. Furthermore, Figures 5.5 and 5.6 represent example OCL for the data quality estimation in the Temperature and Rain Rate measures.

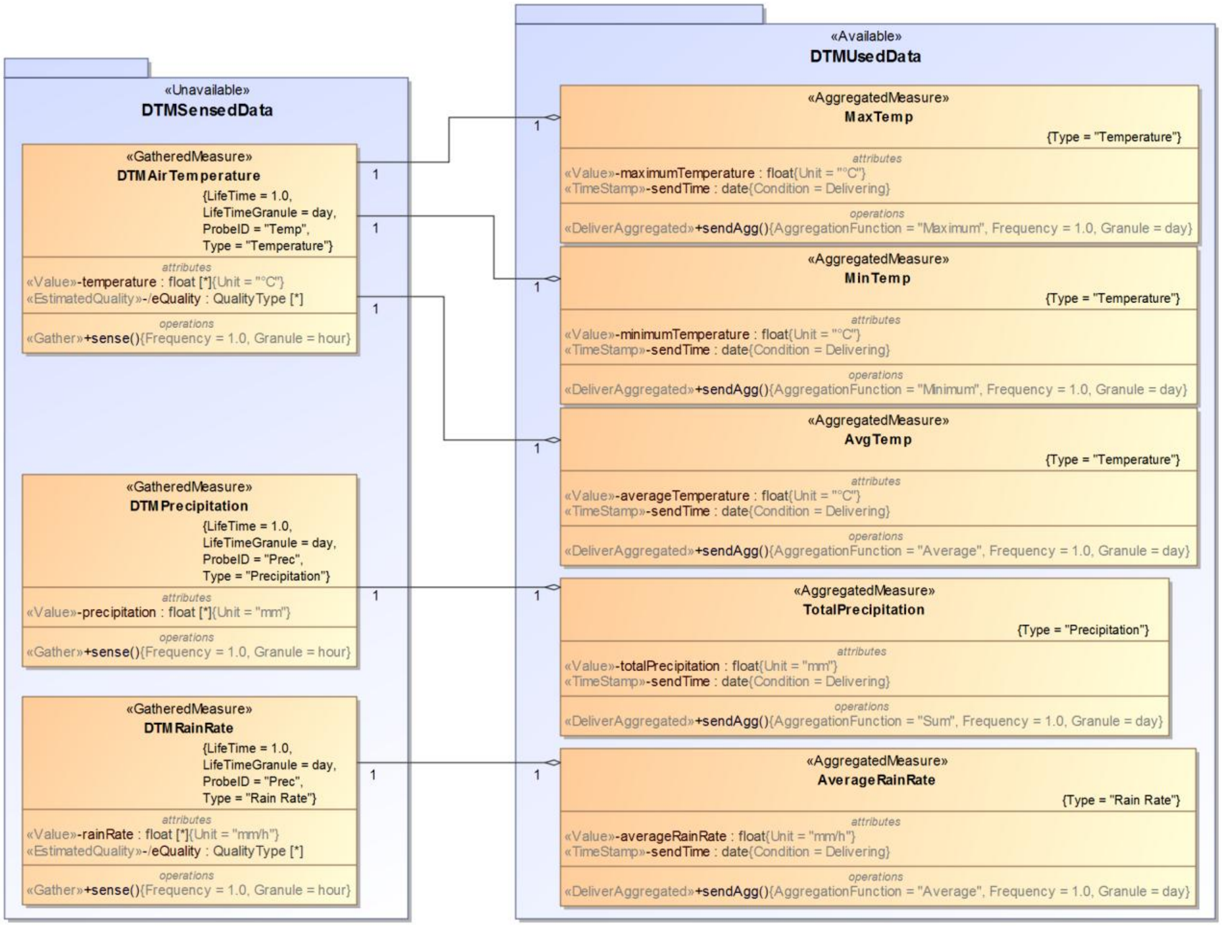

Figure 5.4. Improved UML model for the Aquarisc station data processing

Context	DTMAirTemperature
Name	Temperature quality standard
OCL	((temperature > 0 and temperature < 40) implies eQuality = Good) and ((temperature < 0 or temperature > 40) implies eQuality = Erroneous)
ErrorMessage	If the temperature is between 0°C and 40°C the data quality is Good. Otherwise, the data quality is Erroneous.

Context	MaxTemp
Name	MaxTemperature aggregation standard
OCL	DTMAirTemperature->reject(t|t.eQuality <> Good)
ErrorMessage	Aggregate only temperature measurements with Good eQuality.

Figure 5.5. OCL examples for the estimation and consideration of the data quality in the Temperature measures.

Context	DTMRainRate
Name	RainRate quality standard
OCL	((rainRate > 0 and rainRate < 2500) implies eQuality = Good) and ((rainRate < 0 or rainRate > 2500) implies eQuality = Erroneous)
ErrorMessage	If the rainRate is between 0mm/h and 2500mm/h the data quality is Good. Otherwise, the data quality is Erroneous.

Context	AverageRainRate
Name	RainRate aggregation standard
OCL	DTMRainRate->reject(rr|rr.eQuality <> Good)
ErrorMessage	Aggregate only rainRate measurements with Good eQuality.

Figure 5.6. OCL examples for the estimation and consideration of the data quality in the RainRate measures.

The improved model (Figure 5.4) gathers the meteorological data in the same way that the actual model (Figure 5.3). However, it estimates the data quality of the Temperature and the RainRate according to some logical and contextual boundaries defined in the OCL examples: the temperature must be higher than 0 °C and lower than 40 °C (Figure 5.5), and the rain rate cannot be lower than 0 mm/h or higher than 2500 mm/h (Figure 5.6). Then, it aggregates all the good-quality data with from the three measures: Maximum, Minimum and Average for Temperature, Average for RainRate, and Sum for Precipitation. Finally, it only delivers the daily aggregates to the central server.

The implementation of this improved model (Figure 5.4) in the weather station used by Aquarisc would allow them to receive and store only the most useful, good-quality data. Thus, it reduces the requirements of their central servers in both storage and computing. Indeed, the model and the OCL examples define how the station could estimate and use the data quality for delivering the aggregated measures.

Considering this specific case study, we must state that some properties of our profile are not relevant and that it could have required some extra features not considered in our profile. Indeed, this case study uses a weather station, a hardware platform more complex and powerful than regular WS. Such stations usually rely on multiple energy sources; thus, they do not require to monitor the battery state or to define a sleep state. Besides, they can execute complex calculations with different kinds of measures; a rare feature in common WS that is not considered in our profile. Nevertheless, the specific needs in this case study did not rely on these extra features. Hence, with our UML profile, we could model the data behaviour for its analysis, and even proposed some improvements considering the users' requirements.

5.2.3. Universidad del Cauca - silkworms’ incubator

The silkworms incubator is an important part of a project of the Universidad del Cauca to improve the regional sericulture (*i.e.* silk production) industry [77]. This incubator is a climate chamber that provides the ideal conditions for the silkworms to grow and produce high-quality silk. Hence, the incubator relies on a complex structured monitoring system (*i.e.* a sensor platform) to enable the control of the internal conditions.

The sensor platform of the incubator is based on a conventional Raspberry Pi version 3 (RPi3) hardware platform, which has nine DHT22 humidity and temperature probes, and one APDS-9301 light probe. Since the incubator has nine different levels to place the silkworms, each incubator level has one probe to monitor the specific humidity and temperature conditions of the worms, and the light probe is placed at the middle of the chamber (in level 5) to check the mean luminosity [77].

According to Duque-Torres *et al.* ([77]), the sensor platform gathers data from all the probes every two seconds, stores all the gathered measurements during five minutes, and delivers the data to a storage centre at the end of each five-minute window. However, not all the measurements (about 150 per window) are delivered to the central storage server. Instead, for each five-minute window, the platform calculates and delivers the average temperature and average humidity, and simply delivers the last gathered value of luminosity.

For modelling and validation purposes, we analyse a small data subset that represents the data delivered by the silkworm incubator (Table 5.4).

Table 5.4. Data subset for the analysis of the silkworm incubator

Date	Temperature 0	Humidity 0	Luminosity
20/12/2017 00:00	28,3	83,0	0,0
20/12/2017 00:05	28,0	87,9	0,0
20/12/2017 00:10	28,2	83,0	0,0
20/12/2017 00:15	28,1	87,8	0,0
20/12/2017 00:20	28,3	83,2	0,0

Based on the analysis of this data (Table 5.4), the platform characteristics, and considering our profile, we propose the following UML model for the (shortened) description of the data in the sensor platform of the silkworm incubator (Figure 5.7).

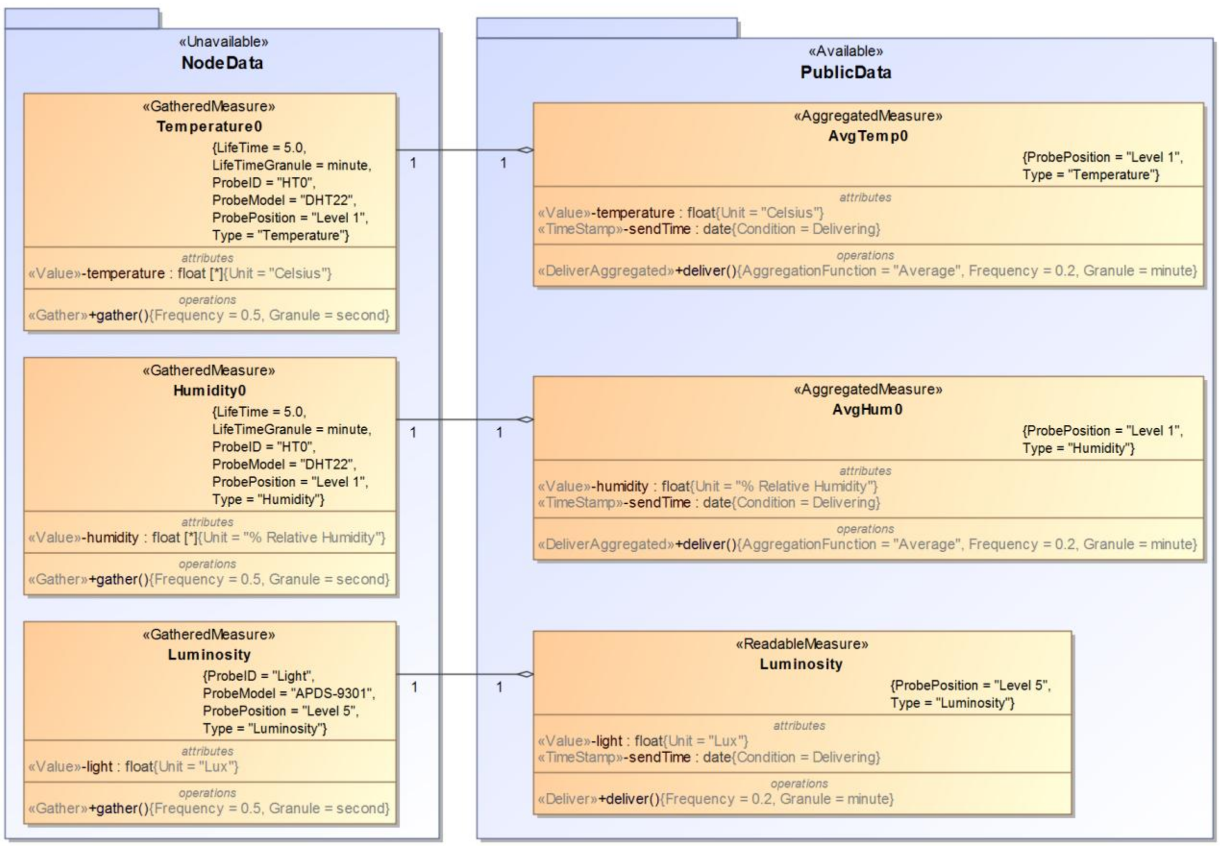

Figure 5.7. Shortened UML model of the incubator data from the user point-of-view.

The modelled node (Figure 5.7) gathers and delivers three types of measurements: Temperature in degrees Celsius (Celsius), Humidity in percentage of Relative Humidity

(% Relative Humidity), and Luminosity in Luxes (Lux). The Humidity (Humidity0) and Temperature (Temperature0) measures are gathered in the first level of the incubator (Level 1), and the measure of Luminosity (Luminosity) is gathered in the fifth or central level of the incubator (Level 5). This model (Figure 5.7) does not include the other Temperature and Humidity measures from the Levels 2 to 9; however, since those measures have the same behaviour than the ones on Level 1 we consider their explanation as redundant. Nevertheless, Figure 5.8 represents the behaviour of all the measures from the incubator.

In the node model (Figure 5.7), it is clear that only the measurement values are being gathered, *i.e.* no other information like battery or time are being considered in the node-internal data. However, the delivering time is considered in all the measures for the node-external data.

Also, it shows that all the gathered measures have the same gathering frequency: 0.5 measurements per second (*i.e.* one measurement every two seconds), without a maximum number of measurements or a working window. However, since the Incubator requires aggregated data (average), some of these gathered measures (the Temperature and Humidity) configure a LifeTime of five minutes, *i.e.* the measured values will be stored inside the node in five-minute windows to calculate the aggregates (average).

Thereby, the model (Figure 5.7) also illustrates that every five minutes (*i.e.* with a frequency of 0.2 values per minute) the node delivers the average value of Temperature and Humidity with their delivery timestamps; and the last gathered value of Luminosity with its delivery timestamp. With these data, the silkworms incubator can control its internal conditions to establish good conditions for the silkworms and improve the produced silk quality [77]. Moreover, the users are able to manually monitor the environment of the silkworms from the central server, checking for undesired conditions that the incubator is not controlling and could affect the insects or the silk production.

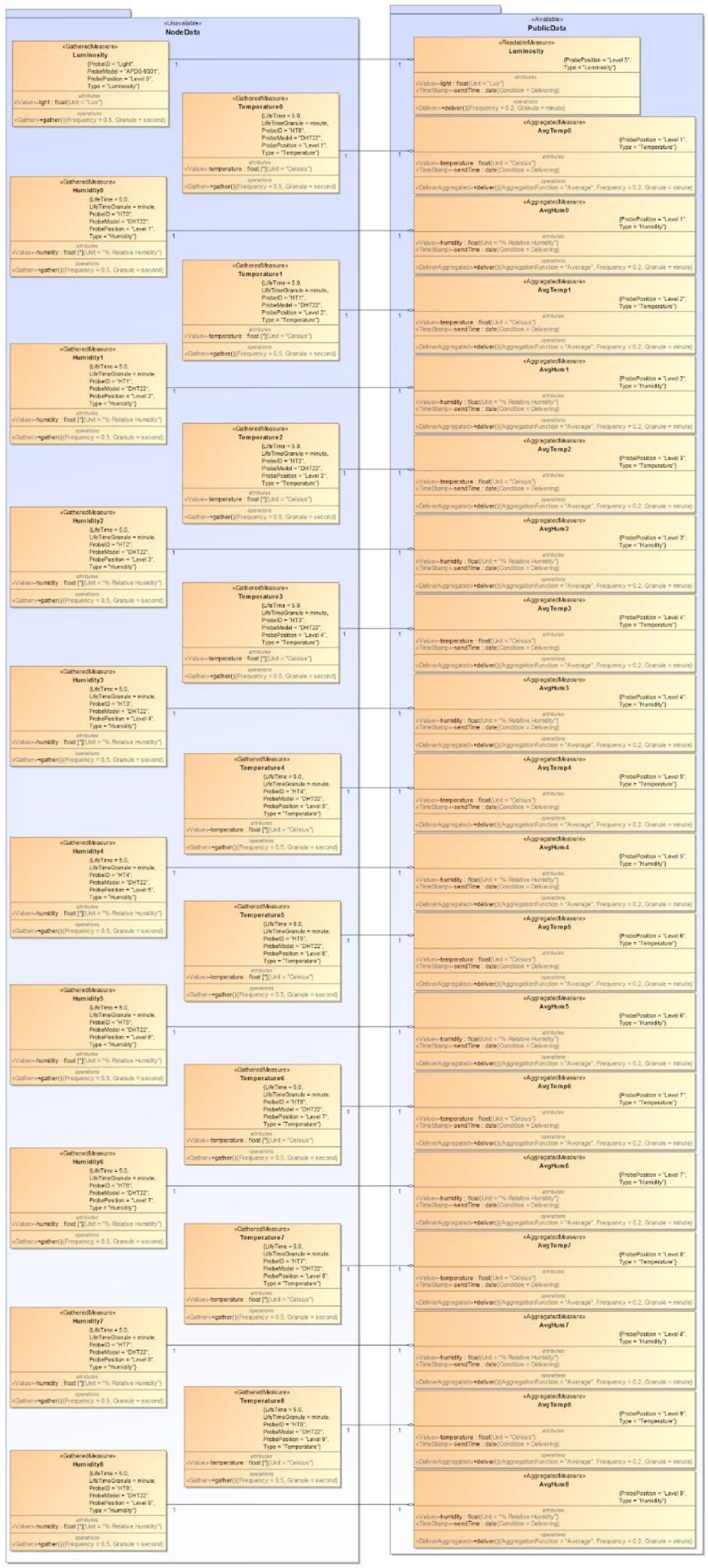

Figure 5.8. Complete model for the Silkworms Incubator Node

This visual model of the incubator node (Figure 5.7) allows the project members ([77]) to formalise their monitoring system. This formalisation is not only useful for writing research reports and papers, and for understanding the delivered data; it is also a powerful tool for the multidisciplinary team to understand their system behaviour without studying the source code. Thus, it could enable a wider discussion among all the members aiming to drive changes in the system, which finally derives in successfully supplying the final users' needs.

Nevertheless, in this particular case study [77], some properties of our UML profile are not considered. For instance, since the incubator requires huge amounts of energy to maintain an ideal environment for the silkworms, it is connected to a dependable (unlimited) power source, for which the energy consumed by the sensor node is negligible. Hence, considering the Battery Level or defining the active periods in operational Windows is irrelevant in this case. Besides, since the Gathering Timestamp is unnecessary in the Average aggregation operation, it is not considered.

Similarly, the data Quality estimation is not considered in this case study; however, we infer that this case has some issues with the data quality [77]. Indeed, the model (Figure 5.7) shows that the node gathers about 150 Luminosity measurements before delivering only the last gathered value with no aggregation, which could mean that the node cannot rely on one measurement due to its quality (*i.e.* the data could be missing). Thereby, the EstimatedQuality property of our profile, with some quality-defining OCL, could reduce the resource waste by limiting the gathered Luminosity measurements to five or ten for delivering only the last good-quality value. Moreover, considering the data quality for calculating the aggregates increases the dependability of the monitoring system. Besides, other aggregates like maximum or minimum could increase the perceived value of the delivered information.

Finally, regarding the results and analysis of the three case studies, we have validated the pragmatism criteria of our data-centric UML profile for agricultural WS (Figure 4.2) in different contexts, with different (commercial and non-commercial) platforms, and different applications. Therefore, our profile is useful for the design and analysis of the query processing in WS applications, being able to replace the real WS for this purpose. Also, after these validation processes, it can be considered as complete for the scope and objectives of this thesis.

5.3. CASE-Tool Validation

After our profile proved that it is interesting and represents an important part of WS through a successful academic validation and that it can design the query processing in WS applications through the design of three real agriculture-oriented case studies, it is complete for the scope and objectives of this thesis. Nevertheless, we must evaluate its correctness and consistency in the UML standard with the CASE tool MagicDraw [10].

We use MagicDraw since it is a CASE tool that allows defining UML profiles with specific meta-model-level and semantic-coherence constraints in OCL for classes and objects. Moreover, it allows using our profile (with stereotypes, tags, and constraints) in the definition of new valid UML models for the WS data behaviour.

This final validation process consists of two steps: firstly, we implement our UML profile in MagicDraw and evaluate if it complies with the language-specific (UML) structure and constraints. Secondly, we define different UML models from our profile (including all the models from the use examples and case studies) to evaluate them in the UML standard and with the profile specific constraints (meta-model level and semantic coherence).

5.3.1. Standard UML evaluation

For this evaluation phase, after implementing our profile and its models in MagicDraw, we used 12 different tests predefined in this CASE tool to check their compliance with the UML standard. For example, these tests allow evaluating the integrity of the profile, the models' composition, their relationships, or even the numbering and spelling. However, the most important tests are the UML Completeness and UML Correctness [78].

The completeness test includes a collection of rules to check if a model is complete, that there are no gaps, and the essential information fields in the elements have been filled in. Besides, the correctness test has a collection of rules that check common mistakes of UML modelling [78]. Indeed, Figure 5.9 shows the selection of the UML Correctness test in MagicDraw, while Figure 5.10 illustrates the results of the test (*i.e.* "Validation Completed - validation was successful."), besides the log of the 12 successful tests.

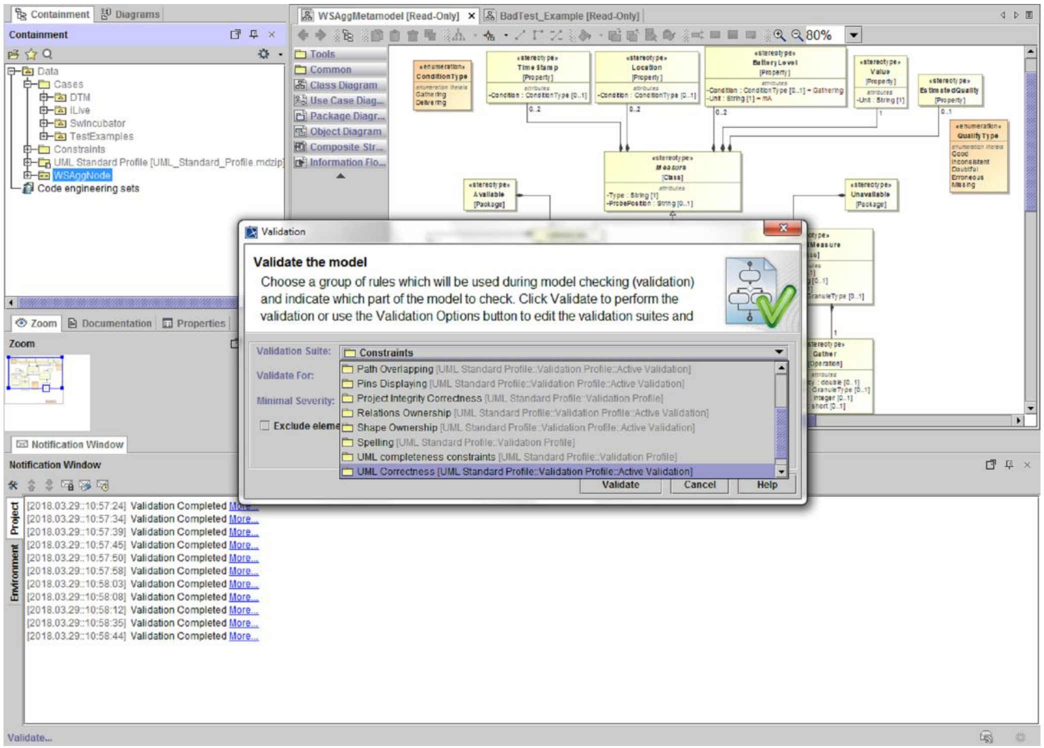

Figure 5.9. UML Correctness test selection

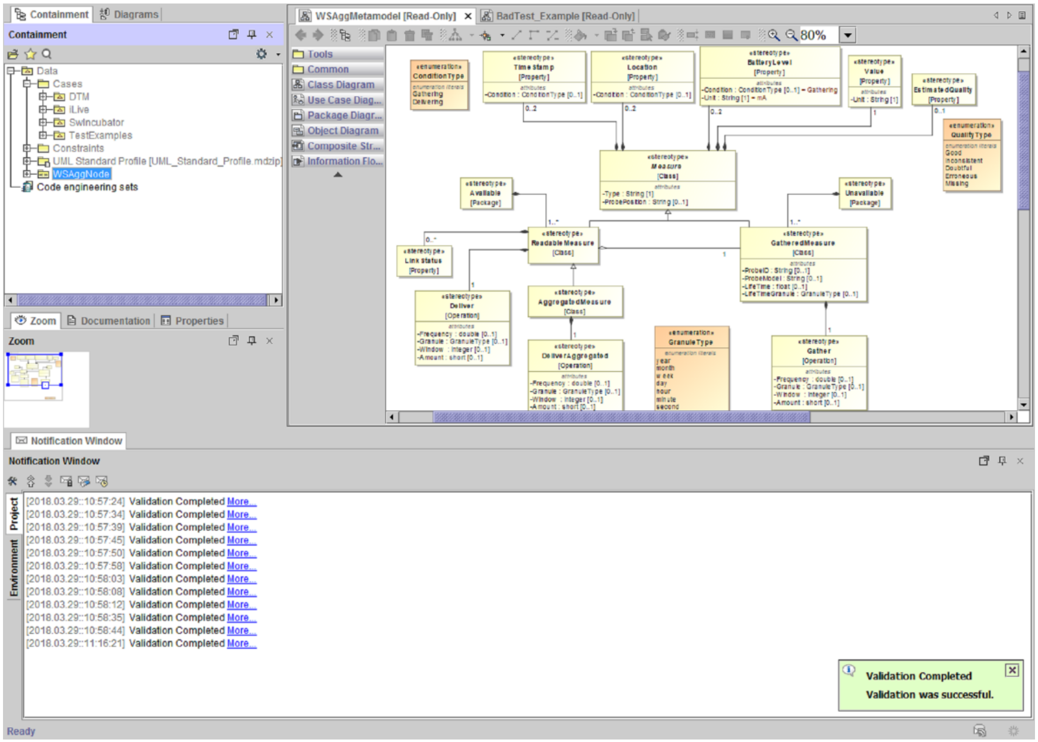

Figure 5.10. UML Correctness test result and results log

The results of these 12 tests (Figures 5.9 and 5.10) evidence that our profile is well-formed in the UML standard; thus, it can be used to design the WS query processing without compatibility problems with the UML2 default profile.

5.3.2. Profile specific evaluation

For this evaluation phase, after successfully evaluating our profile in the UML standard, we define a custom test in MagicDraw to check if the generated models (examples and case studies) comply with the meta-model-level and semantic-coherence constraints of our UML profile. Besides, we also generate a faulty model with different classes implementing various stereotypes of our profile as a control to check if the CASE tool is evaluating the OCL constraints. For instance, this faulty implementation should fail to meet the OCL examples defined in Chapter 4.

Figure 5.11 shows the selection of our custom test (named “Constraints”), which is executed after the standard UML evaluation. However, unlike that evaluation (Figure

5.10), this one presents several errors in the results log (Figure 5.12). All these errors belong to the control group (located in the "**Bad Test**" package), which was intentionally designed to present errors on the basic constraints of the **Measure**, **GatheredMeasure** and **ReadableMeasure** stereotypes; thus, this indicates that the profile correctness checker is working properly.

Figure 5.12 shows the validation results for all the models generated with our UML profile. Since this evaluation presents errors, MagicDraw highlights all the erroneous elements in red boxes, marks all the packages containing erroneous elements with a grey "**X**" in the Containment tab, and lists all the violated constraints in the results log.

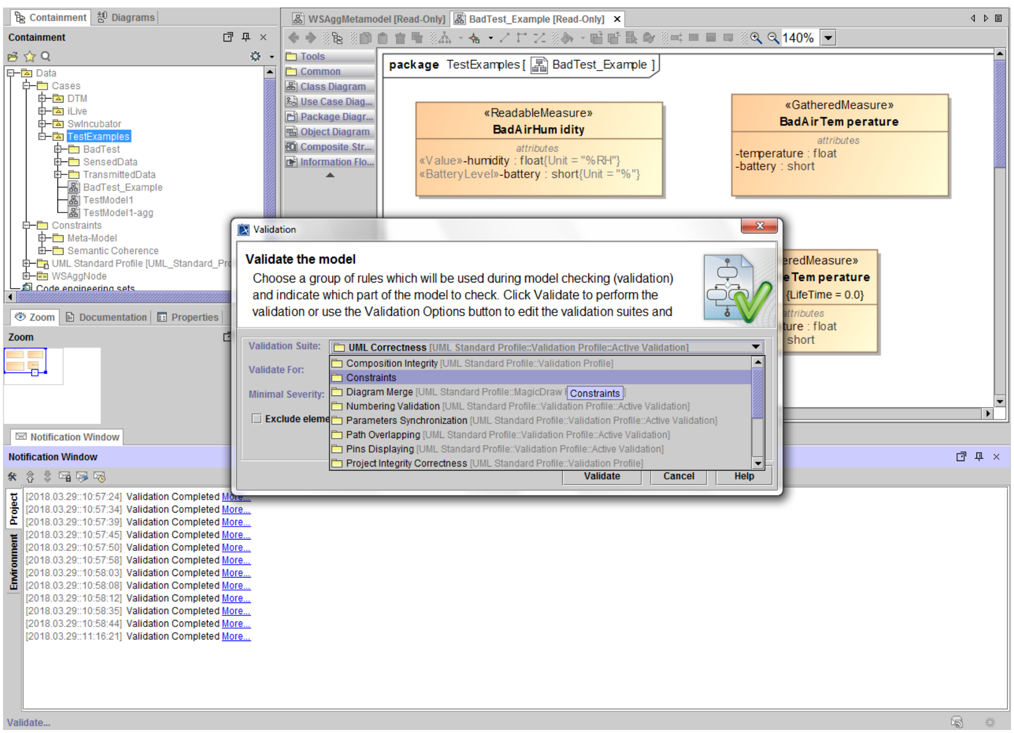

Figure 5.11. Custom test selection

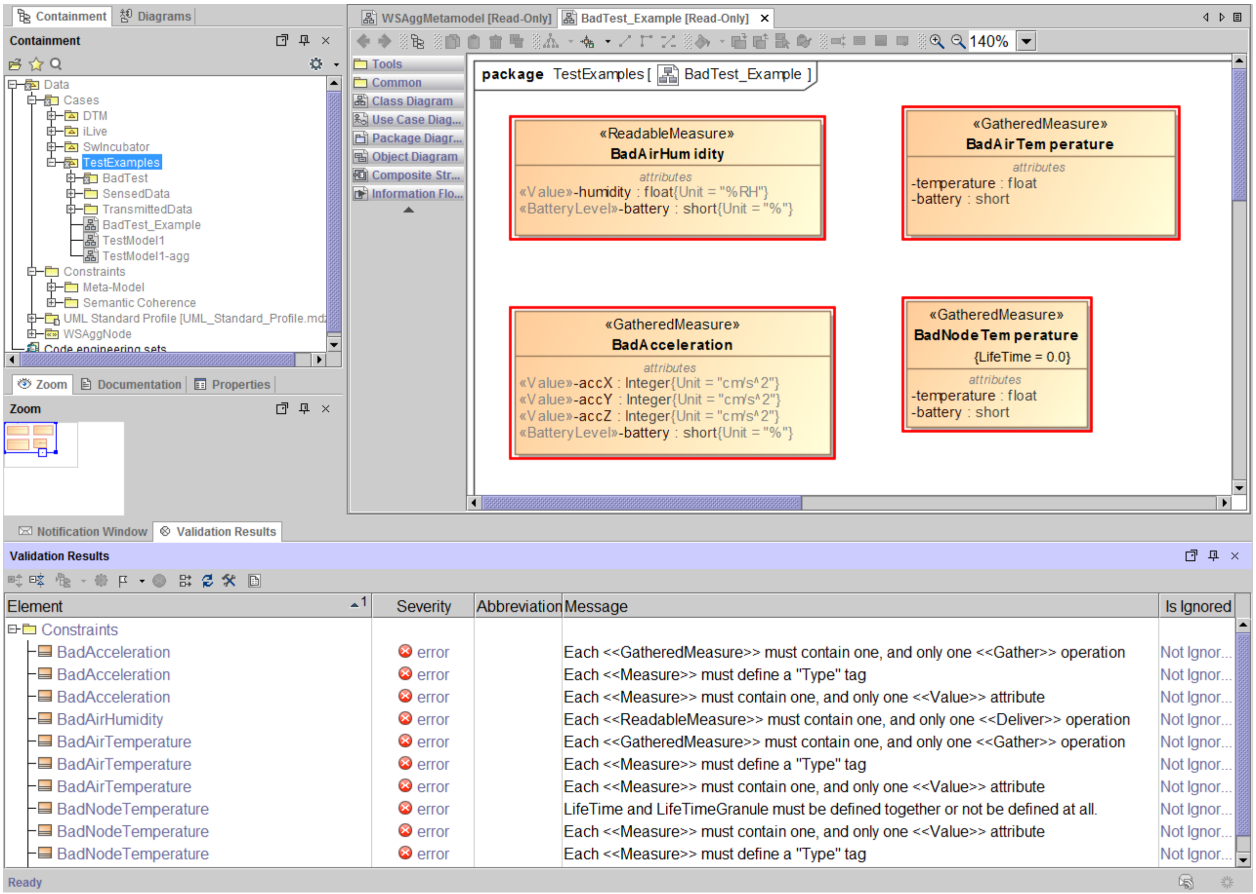

Figure 5.12. Custom test results

These results (Figure 5.12) contain several errors from the faulty (control) model. Indeed, this model violates various constraints of our profile in different ways, *e.g.* defines more than one attribute with the Value stereotype, defines no attributes with the Value stereotype, and does not define any operation using the Gather or Deliver stereotypes. Nevertheless, no other package, from the examples or the case studies, is marked with the grey "**X**", *i.e.* all these data models comply with the profile-specific constraints.

Hence, since our UML profile is well-formed in the UML standard and it allows for the definition and evaluation of new error-free models, providing specific structure and constraints, we can conclude that it is correct and consistent, and that it is a complete meta-model [10], [31].

Summary

In this chapter, we have validated our data-centric UML profile for agricultural WS with three different methods: Academic Validation, a hybrid qualitative and quantitative method for checking with experts if the WS data is represented in our profile. Case Studies, a qualitative method for assessing if our profile can be used to design and analyse the query processing in diverse WS applications. And CASE-Tool Validation, a quantitative method for evaluating the correctness and consistency of our conceptual model with the standard UML constraints [10], [31], [32], [71].

All these validation methods provided positive results about our conceptual model. In the Academic Validation, the experts and practitioners highlighted many advantages of our profile and how it could increase the value of present and future systems, and also stated relevant considerations for future research. In the Case Studies, we identified various problems and possible improvements in the modelled systems. These issues seemed obvious in the analysis of the models; however, they were not identified by studying only the source code of the sensors or the delivered data. Finally, the CASE-Tool Validation with MagicDraw proved the correctness and consistency of our meta-model as a complete UML profile, *i.e.* a lightweight extension of the UML standard that preserves its rules and adds a new specification for the description of the data in sensor platforms.

Therefore, our UML profile can be used as a framework for modelling the query processing in individual Wireless Sensors, which enables the analysis of the system's data behaviour in the design phase, before its implementation.

Chapter 6

Conclusions and Future Works

In this dissertation, we have presented a data-centric UML profile for the design of the data collected and managed by agriculture oriented WS from the user point-of-view. Our profile allows for describing the WS data with different characteristics and configurations, clearly separating the features of the gathered (unavailable) data and the delivered (available) data.

In this chapter, we present the conclusions obtained from the development of this master's thesis, especially considering the results of the validation process. Furthermore, we propose some possible future works that could improve the description of WS and WSN data based on our conceptual model.

6.1. Conclusions

Based on the results of this research, and considering the analysis of our conceptual model, we can conclude that:

- A double mapping process of the state of the art allowed us to identify relevant research on WSN data processing, including the evaluation of quality and the execution of complex operations. Besides, it also highlighted the importance of an accurate data design phase for meeting the user- and application-specific data requirements. However, we found no evidence of design tools (*i.e.* conceptual meta-models) for modelling the WSN data from the user point-of-view.

- A thorough systematic literature review allowed us to discover a complete set of features for the description of the data handled by an agricultural WS. This set of features considers industry and academic proposals for IoT, along with some of the most relevant challenges, characteristics and constraints of agriculture-oriented WSN applications. Hence, it constitutes an initial framework for the description of the data in agricultural WS.

- Though the features discovery process focused on WSN applications for the Agri-food domain, these features could allow for the description of WS data in other contexts since they are based on the IPSO standard guidelines for generic smart sensors [66].

- The process of selecting and grouping the features allowed us to identify the relationships among the features to build a complete model in a simple, formal and organised way. Moreover, this process also enabled the simplification of our meta-model, reducing its original size and complexity, which made it a concise representation of WS [31].

- Since our conceptual meta-model is composed of an organised set of stereotypes, tagged values, and constraints, it can be considered a complete profile in the UML standard.

- Our UML profile is complete enough to be used for modelling different agricultural applications regarding the WS and WSN data behaviour from the user point-of-view.

- Our UML profile provides a complete and effective representation of the WS data behaviour, which could allow for the implementation of model-driven WSN applications that supply the end-user needs.

- A thorough academic validation process with various experts and practitioners in different frameworks evidence that our UML profile is a clear and simple representation of the WS data. Hence, it satisfies the mapping criteria of conceptual models [31].

- The validation of our profile in three real case studies shows that it can be used for describing the data managed by real WS in real WSN applications with different energy configurations in both commercial and non-commercial platforms.

- Our UML profile allows increasing the user-perceived value of the WSN since it provides different features for configuring the node operations, aggregating the gathered data, and checking the data quality.

- Our UML profile is useful for the design and analysis of the query processing in WS applications, being able to replace the real WS for this purpose. Indeed, the conceptual modelling allows for an abstract and direct analysis of the system properties and behaviour in the design, which could improve the effectiveness and efficiency in the implementation [9]. Thereby, our profile satisfies the pragmatism criteria of conceptual models [31].

- The CASE tool implementation of our profile shows its correctness and consistency with the standard UML constraints and the profile-specific OCL. Hence, our profile is well-formed in the UML standard, enabling the definition and evaluation of new error-free models with specifics structures and constraints.

- The models generated with our UML profile help visualise the ideal data behaviour in WS-based systems, specifying their structure and operations, and allowing for the implementation of the real WS and their documentation. Hence, our profile satisfies the set of purposes and benefits of conceptual models [31].

- Since our data-centric UML profile for agricultural WS meets the criteria, principles, and definition of conceptual models and meta-models defined by Rodrigues ([31]) and Jabareen ([32]), it can be considered as complete.

- 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 [12], [58], nor provides complex mechanisms for the execution of multiple data-processing operations in the network [14]. 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.

- Our conceptual 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.

6.2. Future works

The validation process showed various possibilities for improving the performance and usefulness of our profile. Although these improvements are highly relevant, they are out of the scope and objectives of this thesis. Nevertheless, they should be considered to define future related research and as important extensions for this thesis. Thus, we propose the following future works:

- The experimental validation of our UML profile with engineering practitioners to completely prove its usefulness, effectiveness and efficiency. In this experiment, the practitioners must develop a simple WSN application for agriculture, but a group will have access to our profile while the other will be asked to use anything they like. Different variables like perception, success rate, documents' quality, and working time should be considered to compare the results of the profile use.

- The definition of a complete model-to-text mapping mechanism for our profile could allow for automatic code generation to program the WS query processing.

Such mechanism will be very useful for scientists and engineers working on WSN since it eases many tasks in model-driven approaches.

- The definition of relationships between the measures in the UML profile could give our profile a high dynamism, enabling the modelling of more diverse and complex systems with enhanced capabilities. For instance, such relationships could enable the definition of a spatial aggregation mechanism for multiple measurements of the same type gathered from different relative positions by the same node (aggregation between probes with the same type of measure). Moreover, they also allow for the definition of re-aggregation mechanisms for distributive and algebraic operations, which eliminate the memory constraints in temporal aggregation.
- The expansion of our profile with more specific IPSO-compliant sensors (e.g. accelerometer) will ease the representation of these multi-measure variables (though the current profile enables their representation). Moreover, it will make our profile attractive to different areas besides agriculture.
- The expansion of our profile with different types of data analysis besides aggregation could highly increase its usefulness in both agricultural and non-agricultural applications.
- The modelling of the data behaviour in the different levels of a WSN from the individual WS to the clusters and the whole network could allow for complete WSN application representations, enabling the definition of more complex data processing operations, including spatio-temporal aggregation mechanisms for inter-nodes data.
- Finally, since our profile allows for a better standardisation in the design of WSN data, an integrated model including the decision support and data analysis systems like Business Intelligence or Machine Learning could highly increase its impact and effectiveness for the definition of model-driven WSN.

References

[1] V. Rossi, P. Meriggi, T. Caffi, and S. G. and T. Bettati, ‘A Web-based Decision Support System for Managing Durum Wheat Crops’, *Decision Support Systems, Advances in*, 2010.

[2] N. Wang, N. Zhang, and M. Wang, ‘Wireless sensors in agriculture and food industry—Recent development and future perspective’, *Computers and Electronics in Agriculture*, vol. 50, no. 1, pp. 1–14, Jan. 2006.

[3] R. J. Lehmann, R. Reiche, and G. Schiefer, ‘Future internet and the agri-food sector: State-of-the-art in literature and research’, *Computers and Electronics in Agriculture*, vol. 89, pp. 158–174, Nov. 2012.

[4] J. E. Plazas, I. D. López, and J. C. Corrales, ‘A Tool for Classification of Cacao Production in Colombia Based on Multiple Classifier Systems’, in *Computational Science and Its Applications – ICCSA 2017*, 2017, pp. 60–69.

[5] J. E. Plazas and J. C. Corrales, ‘Deploying Timely Alerts Through Converged Services: An Application for Colombian Agriculture’, in *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change*, 2017, pp. 124–138.

[6] F. Jabeen and S. Nawaz, ‘In-network wireless sensor network query processors: State of the art, challenges and future directions’, *Information Fusion*, vol. 25, pp. 1–15, Sep. 2015.

[7] M. H. Anisi, G. Abdul-Salaam, and A. H. Abdullah, ‘A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture’, *Precision Agric*, vol. 16, no. 2, pp. 216–238, Apr. 2015.

[8] Y. Shao, J. Ren, and J. B. Campbell, ‘9.04 - Multitemporal Remote Sensing Data Analysis for Agricultural Application’, in *Comprehensive Remote Sensing*, S. Liang, Ed. Oxford: Elsevier, 2018, pp. 29–38.

[9] J.-R. Abrial, *Modeling in Event-B: system and software engineering*. Cambridge University Press, 2010.

[10] H. Marouane, C. Duvallet, A. Makni, R. Bouaziz, and B. Sadeg, ‘An UML profile for representing real-time design patterns’, *Journal of King Saud University - Computer and Information Sciences*, Jun. 2017.

[11] S. Uke and R. Thool, 'UML Based Modeling for Data Aggregation in Secured Wireless Sensor Network', *Procedia Computer Science*, vol. 78, pp. 706–713, Jan. 2016.
[12] B. Prathiba, K. J. Sankar, and V. Sumalatha, 'Enhancing the data quality in wireless sensor networks #x2014; A review', in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 2016, pp. 448–454.
[13] A. R. Paulon, A. A. Fröhlich, L. B. Becker, and F. P. Basso, 'Wireless sensor network UML profile to support model-driven development', in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 227–232.
[14] N. X. Thang, M. Zapf, and K. Geihs, 'Model Driven Development for Data-centric Sensor Network Applications', in *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, New York, NY, USA, 2011, pp. 194–197.
[15] X. T. Nguyen, H. T. Tran, H. Baraki, and K. Geihs, 'FRASAD: A framework for model-driven IoT Application Development', in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 387–392.
[16] S. Bimonte, M. Schneider, and O. Boussaid, 'Business Intelligence Indicators: Types, Models and Implementation', *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 12, no. 4, pp. 75–98, 2016.
[17] I. Malavolta and H. Muccini, 'A Study on MDE Approaches for Engineering Wireless Sensor Networks', in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2014, pp. 149–157.
[18] F. Essaadi, Y. B. Maissa, and M. Dahchour, 'MDE-Based Languages for Wireless Sensor Networks Modeling: A Systematic Mapping Study', in *Advances in Ubiquitous Networking 2*, Springer, Singapore, 2017, pp. 331–346.
[19] J. Ramirez-Villegas, M. Salazar, A. Jarvis, and C. E. Navarro-Racines, 'A way forward on adaptation to climate change in Colombian agriculture: perspectives towards 2050', *Climatic Change*, vol. 115, no. 3–4, pp. 611–628, May 2012.
[20] Akanksha Sharma, Barbara Arese Lucini, Jan Stryjak, and Sylwia Kechiche, 'Agricultural M2M: a platform for expansion', GSMA Intelligence, Apr. 2015.
[21] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, 'Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review', *Sensors*, vol. 17, no. 8, p. 1781, Aug. 2017.
[22] J. E. Plazas, S. Bimonte, G. De Sousa, and J. C. Corrales, 'Data-Centric UML Profile for Wireless Sensors: Application to Smart Farming', *IJAEIS*, vol. 10, no. 2, 2019.
[23] Aqeel-ur-Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, 'A review of wireless sensors and networks' applications in agriculture', *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 263–270, Feb. 2014.
[24] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, 'Wireless Sensor Networks: A Survey', *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.
[25] L. Atzori, A. Iera, and G. Morabito, 'The Internet of Things: A Survey', *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[26] Patrick Guillemin and Peter Friess, 'Internet of Things: Strategic Research Roadmap', CERP-IoT, Sep. 2009.

[27] J. E. Plazas, J. S. Rojas, D. C. Corrales, and J. C. Corrales, 'Validation of Coffee Rust Warnings Based on Complex Event Processing', in *Computational Science and Its Applications -- ICCSA 2016*, O. Gervasi, B. Murgante, S. Misra, A. M. A. C. Rocha, C. M. Torre, D. Taniar, B. O. Apduhan, E. Stankova, and S. Wang, Eds. Springer International Publishing, 2016, pp. 684–699.

[28] D. C. Corrales, A. Ledezma, A. J. P. Q, J. Hoyos, A. Figueroa, and J. C. Corrales, 'A new dataset for coffee rust detection in Colombian crops base on classifiers', *Sistemas y Telemática*, vol. 12, no. 29, pp. 9–23, Jun. 2014.

[29] G. Pozzani and E. Zimányi, 'Defining Spatio-Temporal Granularities for Raster Data', in *Data Security and Security Data*, 2010, pp. 96–107.

[30] Y. Yao and J. Gehrke, 'The Cougar Approach to In-network Query Processing in Sensor Networks', *SIGMOD Rec.*, vol. 31, no. 3, pp. 9–18, Sep. 2002.

[31] A. Rodrigues da Silva, 'Model-driven engineering: A survey supported by the unified conceptual model', *Computer Languages, Systems & Structures*, vol. 43, pp. 139–155, Oct. 2015.

[32] Y. Jabareen, 'Building a Conceptual Framework: Philosophy, Definitions, and Procedure', *International Journal of Qualitative Methods*, vol. 8, no. 4, pp. 49–62, Dec. 2009.

[33] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, 'Fog Computing and Its Role in the Internet of Things', in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, New York, NY, USA, 2012, pp. 13–16.

[34] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, 'Systematic Mapping Studies in Software Engineering.', in *EASE*, 2008, vol. 8, pp. 68–77.

[35] K. Thramboulidis and F. Christoulakis, 'UML4IoT—A UML-based approach to exploit IoT in cyber-physical manufacturing systems', *Computers in Industry*, vol. 82, no. Supplement C, pp. 259–272, Oct. 2016.

[36] A. Cama-Pinto, G. Pineres-Espitia, R. Zamora-Musa, M. Acosta-Coll, J. Caicedo-Ortiz, and J. Sepúlveda-Ojeda, 'Design of a wireless sensor network for monitoring of flash floods in the city of Barranquilla Colombia', *Ingeniare. Revista Chilena De Ingeniería*, vol. 24, no. 4, pp. 581–599, 2016.

[37] H. Marouane, A. Makni, R. Bouaziz, C. Duvallet, and B. Sadeg, 'Definition of Design Patterns for Advanced Driver Assistance Systems', in *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs*, New York, NY, USA, 2016, p. 3:1–3:10.

[38] A. Luvisi, A. Panattoni, and A. Materazzi, 'RFID temperature sensors for monitoring soil solarization with biodegradable films', *Computers and Electronics in Agriculture*, vol. 123, pp. 135–141, Apr. 2016.

[39] M. Firlej and W. Kresse, 'JAVA-LIBRARY FOR THE ACCESS, STORAGE AND EDITING OF CALIBRATION METADATA OF OPTICAL SENSORS.', *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, 2016.

[40] U. J. Tanik and S. Arkun-Kocadere, 'Cyber-Physical Systems and STEM Development: NASA Digital Astronaut Project', in *Applied Cyber-Physical Systems*, Springer, New York, NY, 2014, pp. 33–49.

[41] G. A. de L. Machado *et al.*, 'Applying Software Engineering Methodology for Designing Biomedical Software Devoted To Electronic Instrumentation', *Journal of Computer Science*, vol. 8, no. 10, pp. 1601–1614, Aug. 2012.
[42] W. Yu, W. W. Dong, and W. R. Feng, 'Designing System for Battlefield Environment Simulation by UML', in *2012 International Conference on Industrial Control and Electronics Engineering*, 2012, pp. 1977–1980.
[43] S. Sicari, L. A. Grieco, G. Boggia, and A. Coen-Porisini, 'DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks', *Journal of Systems and Software*, vol. 85, no. 1, pp. 152–166, Jan. 2012.
[44] S. Meyer, K. Sperner, C. Magerkurth, and J. Pasquier, 'Towards Modeling Real-world Aware Business Processes', in *Proceedings of the Second International Workshop on Web of Things*, New York, NY, USA, 2011, p. 8:1–8:6.
[45] S. Hong, S. Lim, and J. Song, 'Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks: A Survey', *KSII Transactions on Internet and Information Systems*, vol. 5, no. 4, pp. 805–821, Apr. 2011.
[46] Y. Mae, H. Takahashi, K. Ohara, T. Takubo, and T. Arai, 'Component-based robot system design for grasping tasks', *Intel Serv Robotics*, vol. 4, no. 1, pp. 91–98, Jan. 2011.
[47] G. Gu, L. Zhu, Z. Xiong, and H. Ding, 'Design of a Distributed Multiaxis Motion Control System Using the IEEE-1394 Bus', *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4209–4218, Dec. 2010.
[48] H. Kenchannavar, G. S. Patkar, and U. P. Kulakarni, 'Modeling of Coverage Preserving Algorithm Using Object Oriented Methodology with UML', in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, New York, NY, USA, 2010, pp. 708–712.
[49] O. Wongwirat, S. Paelaong, and S. Homchoo, 'ET rescue robot', in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009, pp. 321–321.
[50] S. Jeong, H. Park, J.-E. Lee, S. Lee, and J. Kim, 'The Study of Navigation Software for GNSS Sensor Station', in *26th International Communications Satellite Systems Conference (ICSSC)*, 0 vols, American Institute of Aeronautics and Astronautics, 2008.
[51] V. M. Mohan, 'Advancements in Power System Monitoring and Inter-Operability', PhD Thesis, Citeseer, 2009.
[52] N. Idoudi, C. Duvallet, B. Sadeg, R. Bouaziz, and F. Gargouri, 'Structural Model of Real-Time Databases: An Illustration', in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 58–65.
[53] W. J. Smuda, G. Gerhart, M.-T. Shing, and M. Auguston, 'A software design approach for heterogeneous systems of unattended sensors, unmanned vehicles, and monitoring stations', in *Unmanned/Unattended Sensors and Sensor Networks III*, 2006, vol. 6394, p. 639406.
[54] K. Lee and E. Song, 'A Wireless Environmental Monitoring System Based on the IEEE 1451.1 Standards', in *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, 2006, pp. 1931–1936.

[55] Y. Zhang, N. Chen, J. Wei, and T. Huang, ‘Completing UML Model of Component-Based System with Middleware for Performance Evaluation’, in *Emerging Directions in Embedded and Ubiquitous Computing*, 2006, pp. 72–82.

[56] K. N. Rajanikanth, Y. Narahari, N. N. S. S. R. K. Prasad, and R. S. Rao, ‘A robust and scalable architecture for airborne radar simulation’, in *Conference on Convergent Technologies for Asia-Pacific Region TENCON 2003*, 2003, vol. 1, pp. 173–177 Vol.1.

[57] J. Champeau, P. Dhaussy, R. Moitie, and A. Prigent, ‘Object oriented and formal methods for AUV development’, in *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, 2000, vol. 1, pp. 73–78 vol.1.

[58] G. Jesus, A. Casimiro, and A. Oliveira, ‘A Survey on Data Quality for Dependable Monitoring in Wireless Sensor Networks’, *Sensors*, vol. 17, no. 9, p. 2010, Sep. 2017.

[59] B. Kitchenham and S. Charters, ‘Guidelines for performing systematic literature reviews in software engineering’, in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, School of Computer Science and Mathematics, Keele University: sn, 2007.

[60] J. L. Hatfield and J. H. Prueger, ‘Temperature extremes: Effect on plant growth and development’, *Weather and Climate Extremes*, vol. 10, Part A, pp. 4–10, Dec. 2015.

[61] E. Peres *et al.*, ‘An autonomous intelligent gateway infrastructure for in-field processing in precision viticulture’, *Computers and Electronics in Agriculture*, vol. 78, no. 2, pp. 176–187, Sep. 2011.

[62] W. n Zeng, Y. p Lin, and L. Wang, ‘Privacy-preserving Data Aggregation Based on the P-function Set in Wireless Sensor Networks’, in *2010 10th IEEE International Conference on Computer and Information Technology*, 2010, pp. 2831–2836.

[63] S. M. M. Rahman *et al.*, ‘A Lightweight Secure Data Aggregation Technique for Wireless Sensor Network’, in *2014 IEEE International Symposium on Multimedia*, 2014, pp. 387–392.

[64] Y. Kim, P. Bae, J. Han, and Y.-B. Ko, ‘Data aggregation in precision agriculture for low-power and lossy networks’, in *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2015, pp. 438–443.

[65] C. R. Panigrahi, B. Pati, and J. L. Sarkar, ‘CPDA: A conflict-free periodic data aggregation technique in wireless sensor networks’, *Egyptian Informatics Journal*, Sep. 2016.

[66] ‘Smart Object Interoperability’, *IPSO Alliance*. .

[67] S. Luján-Mora, J. Trujillo, and I.-Y. Song, ‘A UML profile for multidimensional modeling in data warehouses’, *Data & Knowledge Engineering*, vol. 59, no. 3, pp. 725–769, Dec. 2006.

[68] OMG, ‘About the Unified Modeling Language Specification Version 2.4’, Mar-2011. [Online]. Available: http://www.omg.org/spec/UML/2.4/. [Accessed: 04-Mar-2018].

[69] F. Y. V. Cantero, J. C. Corrales, and G. A. R. Gonzalez, ‘Aproximación a un modelo contextual para calidad de datos en agricultura de precisión’, *Revista Ingenierías Universidad de Medellín*, vol. 15, no. 29, pp. 99–112, May 2016.
[70] K. Zhang, Q. Han, Z. Cai, and G. Yin, ‘RiPPAS: A Ring-Based Privacy-Preserving Aggregation Scheme in Wireless Sensor Networks’, *Sensors*, vol. 17, no. 2, p. 300, Feb. 2017.
[71] B. Kitchenham, S. Linkman, and D. Law, ‘DESMET: a methodology for evaluating software engineering methods and tools’, *Computing Control Engineering Journal*, vol. 8, no. 3, pp. 120–126, Jun. 1997.
[72] N. V. N. Chism, E. Douglas, and W. J. Hilson Jr, ‘Qualitative research basics: A guide for engineering educators’, *Rigorous Research in Engineering Education NSF DUE-0341127*, 2008.
[73] X. Liu, K. M. Hou, H. Shi, and C. Guo, ‘Efficient middleware for user-friendly wireless sensor network integrated development environment’, in *2012 Wireless Advanced (WiAd)*, 2012, pp. 22–28.
[74] Irstea, ‘AgroTechnoPôle Auvergne Rhône-Alpes: the creation of an unprecedented structure for innovation in Europe | Irstea’, 26-Feb-2016. [Online]. Available: http://www.irstea.fr/en/all-news/institute/agrotechnopole-auvergne-rhone-alpes-creation-unprecedented-structure-innovation. [Accessed: 04-Mar-2018].
[75] ‘Aquarisc - Home’, 2018. [Online]. Available: http://www.aquarisc.com/. [Accessed: 20-Jul-2018].
[76] C. Valencia-Payan and J. C. Corrales, ‘A Multiscale Based Rainfall Amount Prediction Using Multiple Classifier System’, in *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change*, 2017, pp. 16–28.
[77] A. Duque-Torres *et al.*, ‘A new environmental monitoring system for silkworm incubators’, *F1000Research*, vol. 7, p. 248, Feb. 2018.
[78] No Magic, Inc., *MagicDraw User Manual*, 18.1. 2015.

Appendix A

Model to Text Algorithm

In this Appendix, we present an initial, untested prototype of a mapping algorithm for our data-centric UML profile for agricultural WS, which could allow for the automatic implementation of models developed with our profile into real JSON-supporting sensor nodes. Moreover, we provide two implementation examples based on the hypothetical case studies in Chapter 4.

A.1. Algorithm Definition

In this section, we highlight some characteristics of our Data-centric Wireless-Sensor UML Profile in order to define a protocol for mapping the designed models into a JSON configuration file for its implementation in the nodes.

- Each model will represent the data of one sensor node.
- One node could have more than one measure; which may have the same or different types.
- Each measure is divided in two related classes: gathered (unavailable) and readable (available).

- There is a direct relation between the elements of the available and unavailable classes of a measure. Thus, there will be common elements (gathered and readable) and elements that only belong to one of the two classes.
- The elements of a class are its tags, attributes, and operations.

Sensor nodes are usually configured by means of a text or JSON file to set up the basic data characteristics and configuration (*e.g.* type and unit of measurement, frequency of acquisition and sending, etc.). Therefore, in this subsection we present the mapping of our UML profile towards a JSON configuration file for the sensor platform (Figure A.1).

The algorithm that maps the UML profile element to a JSON configuration file is described as follows:

A.1.1. Mapping algorithm

1. Create an empty JSON file for the modelled WS.
2. Create an object with the name of a measure (*i.e.* <Variable_Name>: {}).
3. Directly map the common tags of the measure into its object. (*i.e.* <Tag_Name>: <Tag_Value>).
4. Create an object into the measure called “attributes”, which will contain all the common attributes of the measure (*i.e.* “attributes”: {}).
    1. Inside the “attributes” object, map a common attribute of the measure like a new object, identified by the name of the attribute and containing the attribute metadata: stereotype, type, cardinality, and other attribute-specific tags (*i.e.* <Attribute_Name>: {“stereotype”: <Stereotype>, “type”: <Type>, “cardinality”: <Cardinality>, <Tag_Name>: <Tag_Value>}).
    2. Repeat the previous sub step (4.a) until all the common attributes of the measure are mapped.
5. Create an object into the measure called “gathered”, which will contain all the unavailable elements of the measure (*i.e.* “gathered”: {}).
6. Directly map the unavailable tags of the measure into the “gathered” object. (*i.e.* <Una_Tag_Name>: <Una_Tag_Value>).

7. Into the "gathered" object, create a new object called "attributes", which will contain all the unavailable attributes of the measure (*i.e.* "attributes": {}).
    1. Inside this "attributes" object, map an unavailable attribute of the measure like a new object, identified by the name of the attribute and containing the attribute metadata: stereotype, type, cardinality, and other attribute-specific tags (*i.e.* <Una_Attribute_Name>: {"stereotype": <Stereotype>, "type": <Type>, "cardinality": <Cardinality>, <Tag_Name>: <Tag_Value>}).
    2. Repeat the previous sub step (7.a) until all the unavailable attributes of the measure are mapped.
8. Into the "gathered" object, create a new object called "operations", which will contain all the unavailable operations of the measure (*i.e.* "operations": {}).
    1. Inside this "operations" object, map an unavailable operation of the measure like a new object, identified by the name of the operation and containing the operation metadata: stereotype and other operation-specific tags (*i.e.* <Una_Operation_Name>: {"stereotype" : <Stereotype>, <Tag_Name> : <Tag_Value>}).
    2. Repeat the previous sub step (8.a) until all the unavailable attributes of the measure are mapped.
9. Create an object into the measure called "readable", which will contain all the available elements of the measure (*i.e.* "readable": {}).
10. Directly map the available tags of the measure into the "readable" object. (*i.e.* <Ava_Tag_Name>: <Ava_Tag_Value>).
11. Into the "readable" object, create a new object called "attributes", which will contain all the available attributes of the measure (*i.e.* "attributes": {}).
    1. Inside this "attributes" object, map an available attribute of the measure like a new object, identified by the name of the attribute and containing the attribute metadata: stereotype, type, cardinality, and other attribute-specific tags (*i.e.* <Ava_Attribute_Name>: {"stereotype": <Stereotype>, "type": <Type>, "cardinality": <Cardinality>, <Tag_Name>: <Tag_Value>}).
    2. Repeat the previous sub step (11.a) until all the available attributes of the measure are mapped.
12. Into the "readable" object, create a new object called "operations", which will contain all the available operations of the measure (*i.e.* "operations": {}).

1. Inside this “operations” object, map an available operation of the measure like a new object, identified by the name of the operation and containing the operation metadata: stereotype and other operation-specific tags (*i.e.* <Ava_Operation_Name>: {“stereotype” : <Stereotype>, <Tag_Name> : <Tag_Value>}).
2. Repeat the previous sub step (12.a) until all the available attributes of the measure are mapped.

13. Finally, repeat steps 2 to 12 until all the measures are mapped in the JSON.

```
{
   <Measure_Name>:{
      <Tag_Name>:<Tag_Value>,
      "attributes":{
       <Attribute_Name>:{
              "stereotype":<Stereotype>,
              "type":<Type>,
              "cardinality":<Cardinality>,
              <Tag_Name>:<Tag_Value>
       }
      },
      "gathered":{
       <Una_Tag_Name>:<Una_Tag_Value>,
       "attributes":{
              <Una_Attribute_Name>:{
                 "stereotype":<Stereotype>,
                 "type":<Type>,
                 "cardinality":<Cardinality>,
                 <Tag_Name>:<Tag_Value>
              }
       },
       "operations":{
              <Una_Operation_Name>:{
                 "stereotype":<Stereotype>,
                 <Tag_Name>:<Tag_Value>
              }
       }
      },
      "readable":{
       <Ava_Tag_Name>:<Ava_Tag_Value>,
       "attributes":{
              <Ava_Attribute_Name>:{
                 "stereotype":<Stereotype>,
                 "type":<Type>,
                 "cardinality":<Cardinality>,
                 <Tag_Name>:<Tag_Value>
              }
       },
       "operations":{
              <Ava_Operation_Name>:{
                 "stereotype":<Stereotype>,
                 <Tag_Name>:<Tag_Value>
              }
       }
```

```
            }
        }
    }
```

Figure A.1. JSON format for the data-centric WS UML profile.

A.2. Examples

In this section, we present two implementation examples for our model to text algorithm, which could map the hypothetical implementations of Chapter 4 (examples 3 and 5 in Section 4.3) into JSON documents.

A.2.1. Example 1

In the third example of Section 4.3 (modelled in Figure 4.5) the hypothetical user needs to know the soil moisture in order to decide if irrigation is needed; expecting to receive no more than three inconsistent or better-quality information about the soil moisture per hour. An implementation of this WS model in a real node could be achieved with a JSON configuration file like Figure A.2.

```
{
   "SoilMoisture":{
      "Type":"Soil Moisture",
      "ProbePosition":"15 cm",
      "attributes":{
       "moisture":{
              "stereotype":"Value",
              "type":"double",
              "cardinality":1,
              "Unit":"VWC"
       },
       "time":{
              "stereotype":"TimeStamp",
              "type":"date",
              "cardinality":1
       },
       "eQuality":{
              "stereotype":"EstimatedQuality",
              "type":"QualityType",
              "cardinality":1
       },
       "battery":{
              "stereotype":"BatteryLevel",
              "type":"float",
              "cardinality":1,
              "Unit":"V"
       }
      },
      "gathered":{
       "ProbeID":"sm01",
       "ProbeModel":"EC-5",
       "operations":{
              "sense":{
                 "stereotype":"Gather",
                 "Frequency":0.1,
                 "Granule":"minute"
              }
       }
      },
      "readable":{
       "attributes":{
              "sendTime":{
                 "stereotype":"TimeStamp",
                 "type":"date",
                 "cardinality":1,
                 "Condition":"Delivering"
              }
       },
       "operations":{
              "send":{
                 "stereotype":"Deliver",
                 "Frequency":0.1,
                 "Granule":"minute",
                 "Window":60,
                 "Amount":3
              }
       }
      }
   }
}
```


Figure A.2. JSON configuration file for example model of a moisture WSN without aggregation.

A.2.2. Example 2

In the fifth example of Section 4.3 (modelled in Figure 4.7) the hypothetical user needs to know when the soil of the crops is too dry in order to irrigate; expecting only good quality information about the minimum soil moisture once per hour. An implementation of this WS model in a real node could be achieved with a JSON configuration file like Figure A.3.


```
{
   "SoilMoisture":{
      "Type":"Soil Moisture",
      "ProbePosition":"15 cm",
      "gathered":{
         "ProbeID":"sm01",
         "ProbeModel":"EC-5",
       "LifeTime":1,
         "LifeTimeGranule":"hour",
       "attributes":{
               "moisture":{
                "stereotype":"Value",
                "type":"double",
               "cardinality":6,
                "Unit":"VWC"
               },
               "time":{
                "stereotype":"TimeStamp",
                "type":"date",
               "cardinality":6
               },
               "eQuality":{
                "stereotype":"EstimatedQuality",
                "type":"QualityType",
               "cardinality":6
               },
               "battery":{
                "stereotype":"BatteryLevel",
                "type":"float",
               "cardinality":6,
               "Unit":"V"
               }
       },
       "operations":{
               "sense":{
                "stereotype":"Gather",
               "Frequency":0.1,
                "Granule":"minute"
               }
       }
      },
      "readable":{
       "attributes":{
               "minMoisture":{
                "stereotype":"Value",
                "type":"double",
               "cardinality":1,
                "Unit":"VWC"
               },
               "senseTime":{
                "stereotype":"TimeStamp",
                "type":"date",
               "cardinality":1,
```

```
                "Condition":"Gathering"
                },
                "sendTime":{
                 "stereotype":"TimeStamp",
                 "type":"date",
                "cardinality":1,
                 "Condition":"Delivering"
                }
        },
        "operations":{
                "sendAgg":{
                 "stereotype":"DeliverAggregated",
                "Frequency":1,
                 "Granule":"hour",
                 "AggregationFunction":"Min"
                }
        }
       }
    }
}
```

Figure A.3. JSON configuration file for example model of a moisture WSN with aggregation.

Appendix B

Paper

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

This is a pre-copy-edited version of the paper entitled "Data-Centric UML Profile for Wireless Sensors: Application to Smart Farming", which has been accepted for publication in the International Journal of Agricultural and Environmental Information Systems (IJAEIS), Volume 10 - Issue 2; indexed in the Web of Science ® (ESCI) and Scopus ® (SJR Q4).

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

Authors
Plazas, Julián Eduardo - Universidad del Cauca, GIT - Popayán, Colombia (jeplazas@unicauca.edu.co)*
Bimonte, Sandro - Irstea, UR TSCF - Aubière, France (sandro.bimonte@irstea.fr)
De Sousa, Gil - Irstea, UR TSCF - Aubière, France (gil.de-sousa@irstea.fr)
Corrales, Juan Carlos - Universidad del Cauca, GIT - Popayán, Colombia (jcorral@unicauca.edu.co)

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, we propose 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, we validate our UML profile through a CASE (Computer-Aided Software Engineering) tool implementation and one case study centred on the data collected by a real WSN implementation for precision agriculture and smart farming.

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

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 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 paper 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 (Patrick Guillemin & Peter 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 signalling 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 metamodels (*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.

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 Modelling 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

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 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.

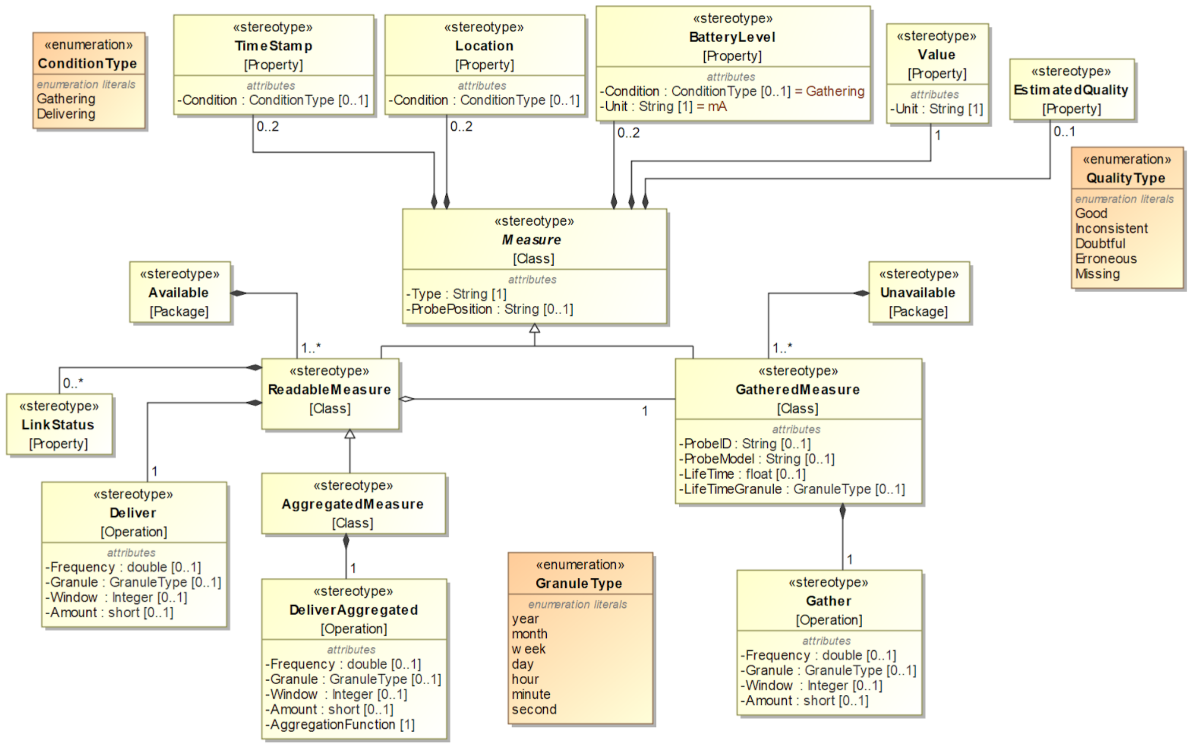

Figure 1. Data-centric Wireless-Sensor UML profile from the user point-of-view.

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.
- 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.

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).

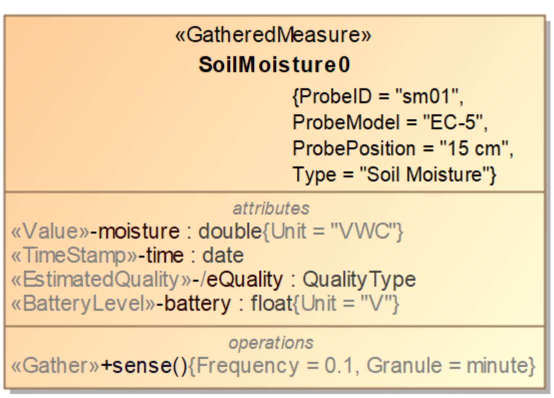

Figure 2. Example Class implementing the **GatheredMeasure** stereotype

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.

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

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 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.

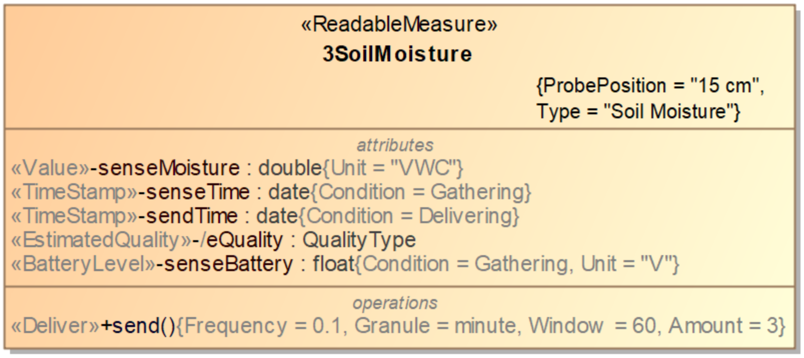

Figure 3. Example Class implementing the **ReadableMeasure** stereotype

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.

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 hour (yellow lines of Table 4), only three values are sent (yellow lines of Table 5).

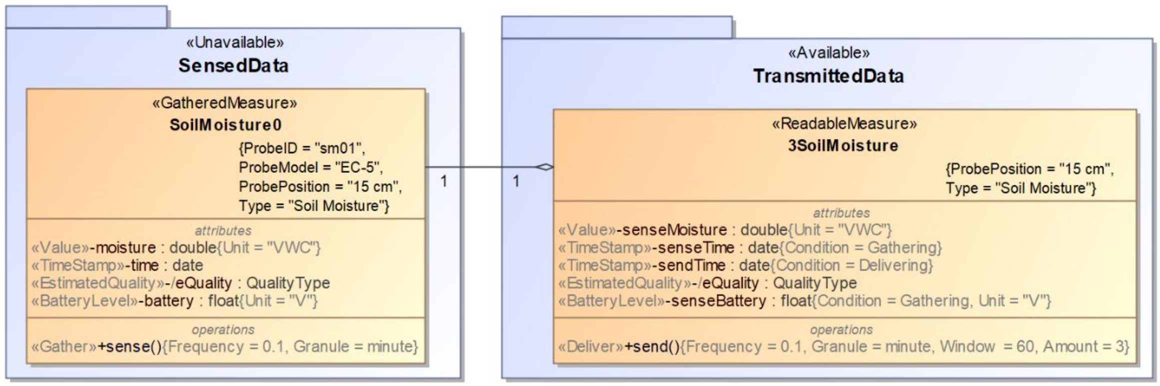

Figure 4. UML model of a moisture WS data from the user point-of-view.

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.

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

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

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 (K. 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 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.

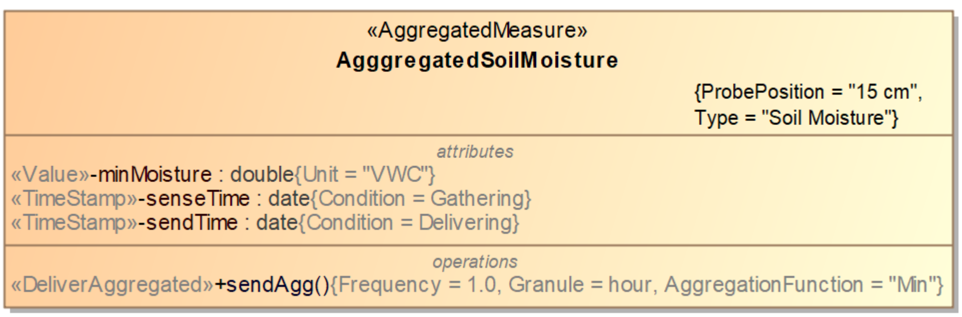

Figure 5. Example Class implementing the **AggregatedMeasure** stereotype

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.

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

Example 5

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.

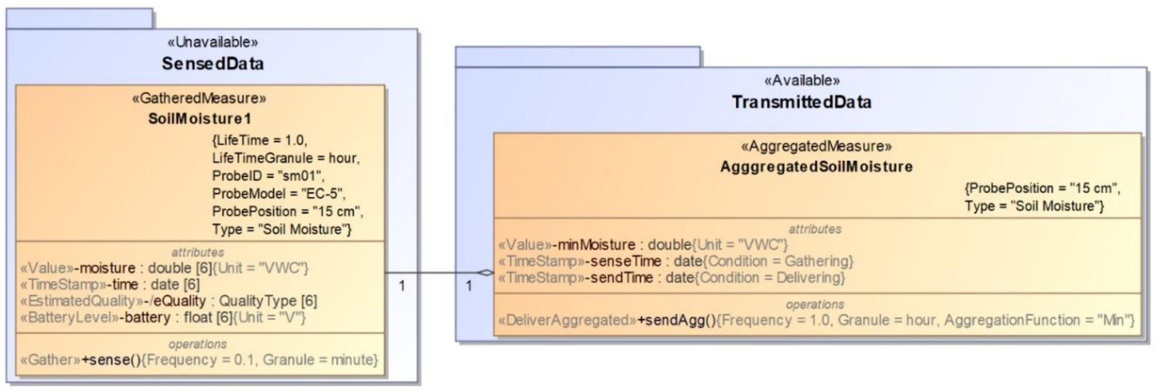

Figure 6. UML model of a moisture WS data with aggregation from the user point-of-view.

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, 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 (yellow lines of Table 7) only the one with the minimum value (40) and Good quality is sent (yellow line of Table 8).

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).

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
…	…		…	…

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
…	…		…	

Profile constraints

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

- *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 (ΣSM) 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).

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

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

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).

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

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.

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

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

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 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).

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 10. OCL application-specific constraints for example 3.

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 to deliver only good-quality data. Thus, it needs to identify the quality of the data and include only good-quality values for aggregation (Figure 11).

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.

Figure 11. OCL application-specific constraints for example 5.

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 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.

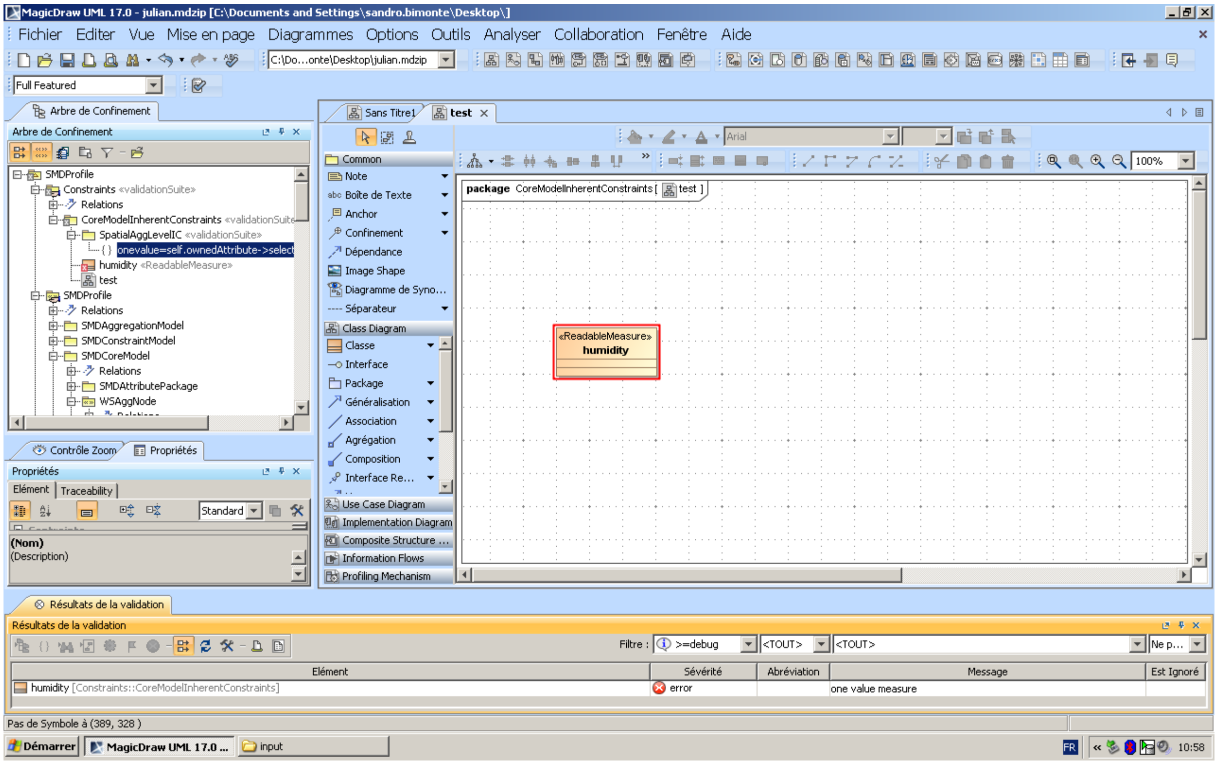

Figure 12. MagicDraw implementation of our profile with OCL constraints

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 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 Irrometer Watermark probes; though not all these latter probes are connected to the nodes. For example, in this experimentation, nodes are only equipped with three Irrometer 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).

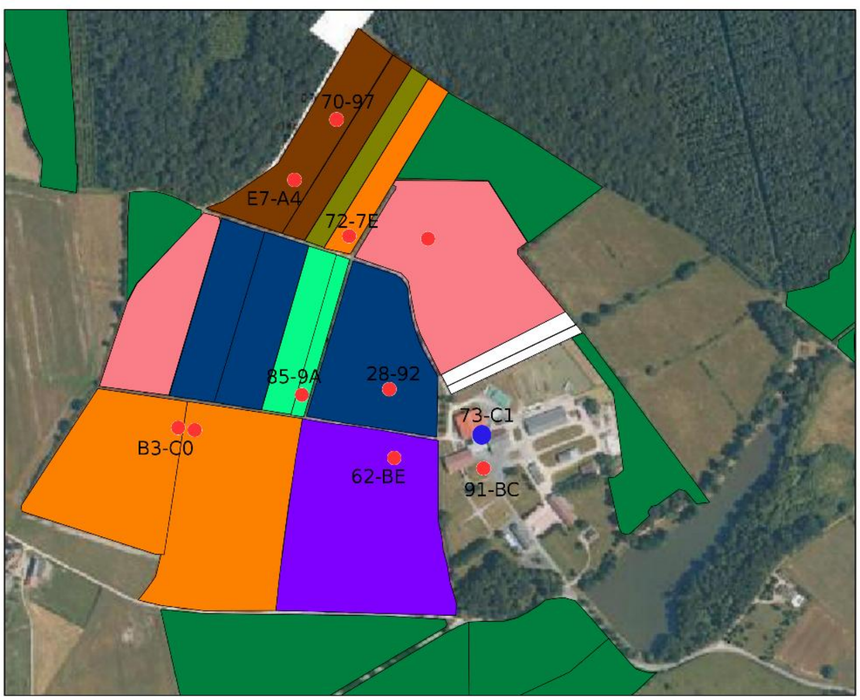

Figure 13. Deployment of the iLive network in the Montoldre site.

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).

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**

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).

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.

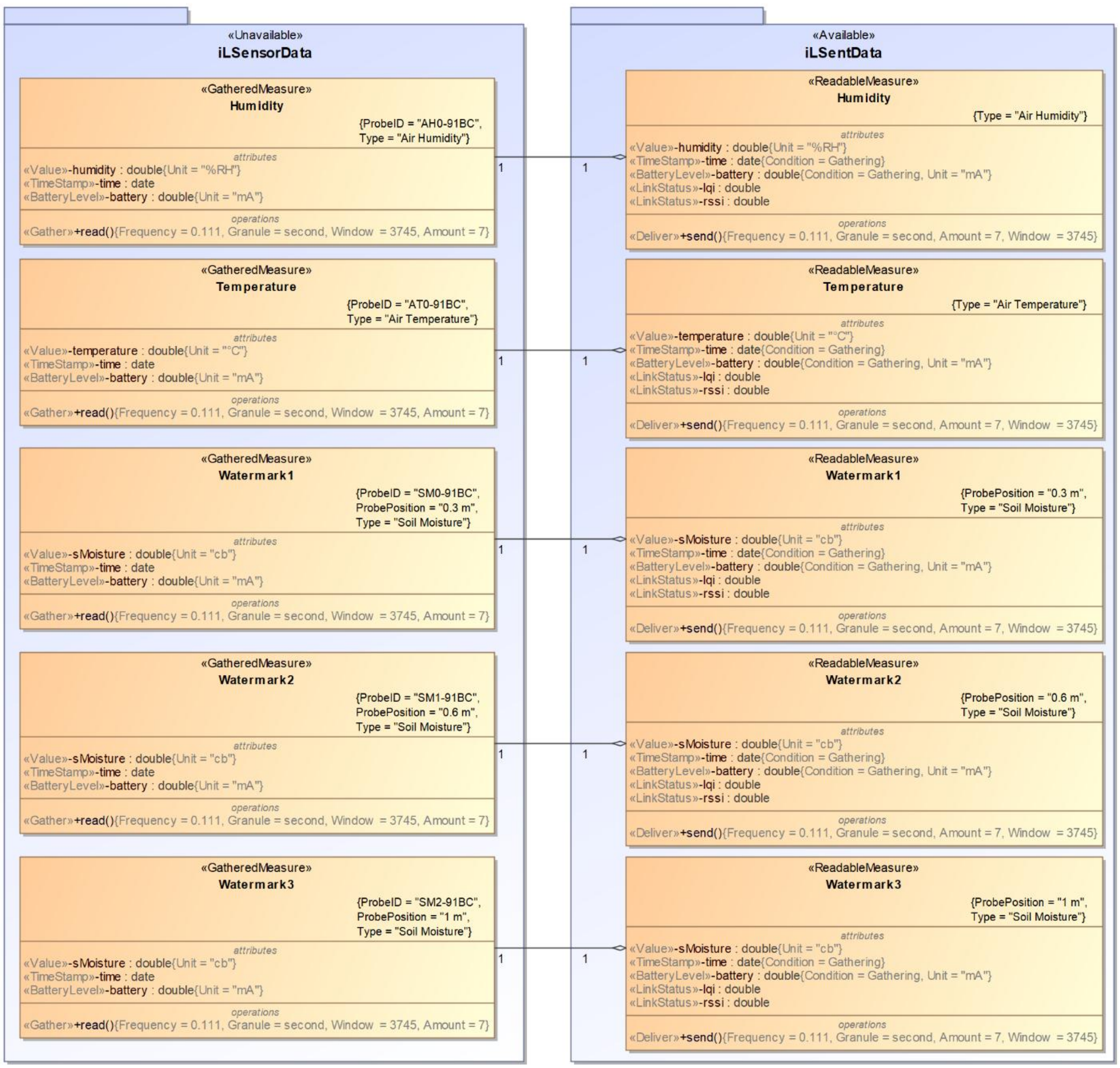

Figure 14. UML data model from the user point-of-view for the iLive case study, node 91-BC.

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 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.

CONCLUSIONS 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 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.

Acknowledgements: 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, Innovacción 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-3848.

REFERENCES

Abrial, J.-R. (2010). *Modeling in Event-B: system and software engineering*. Cambridge University Press.

Akanksha Sharma, Barbara Arese Lucini, Jan Stryjak, & Sylwia Kechiche. (2015). *Agricultural M2M: a platform for expansion*. GSMA Intelligence. Retrieved from https://www.gsmaintelligence.com/research/2015/03/agricultural-m2m-a-platform-for-expansion/479/

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Comput. Netw.*, *38*(4), 393–422. https://doi.org/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. https://doi.org/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. https://doi.org/10.1016/j.csi.2011.03.004

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. *Comput. Netw.*, *54*(15), 2787–2805. https://doi.org/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 (IJDWM)*, *12*(4), 75–98. https://doi.org/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, USA: ACM. https://doi.org/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.

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. https://doi.org/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 (Cat. No.00CH37158)* (Vol. 1, pp. 73–78 vol.1). https://doi.org/10.1109/OCEANS.2000.881236

Firlej, M., & Kresse, W. (2016). JAVA-LIBRARY FOR THE ACCESS, STORAGE AND EDITING OF CALIBRATION METADATA OF OPTICAL SENSORS. *International Archives of the Photogrammetry, Remote Sensing & 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. https://doi.org/10.1109/TIE.2010.2044127

Hong, S., Lim, S., & Song, J. (2011). Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks: A Survey. *KSII Transactions on Internet and Information Systems*, *5*(4), 805–821.

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). https://doi.org/10.1109/ISORC.2008.80

Irstea. (2016, February 26). AgroTechnoPôle Auvergne Rhône-Alpes: the creation of an unprecedented structure for innovation in Europe | Irstea. Retrieved 4 March 2018, 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. https://doi.org/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)* (Vols 1–0). American Institute of Aeronautics and Astronautics. https://doi.org/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*, *17*(9), 2010. https://doi.org/10.3390/s17092010

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, USA: ACM. https://doi.org/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). https://doi.org/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. https://doi.org/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). https://doi.org/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. https://doi.org/10.1016/j.compag.2016.02.023

Machado, G. A. de L., Zacaro, P. M. D., Junior, A. R. de P., Souza, de O. e, Lopes, M., Machado, G. A. de L., … de Oliveira e Souza, M. L. (2012). Applying Software Engineering Methodology for Designing Biomedical Software Devoted To Electronic Instrumentation. *Journal of Computer Science*, *8*(10), 1601–1614. https://doi.org/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. https://doi.org/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*. https://doi.org/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, USA: ACM. https://doi.org/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, USA: ACM. https://doi.org/10.1145/1993966.1993978

Mohan, V. M. (2009). *Advancements in Power System Monitoring and Inter-Operability* (PhD Thesis). Citeseer.

OMG. (2011, March). About the Unified Modeling Language Specification Version 2.4. Retrieved 4 March 2018, from http://www.omg.org/spec/UML/2.4/

Patrick Guillemin, & Peter Friess. (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

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). https://doi.org/10.1109/INDIN.2014.6945512

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *EASE* (Vol. 8, pp. 68–77).

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). Springer, Cham. https://doi.org/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). Springer, Cham. https://doi.org/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. https://doi.org/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, Berlin, Heidelberg. https://doi.org/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). https://doi.org/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, p. 173–177 Vol.1). https://doi.org/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. https://doi.org/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. https://doi.org/10.1016/B978-0-12-409548-9.10431-2

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. https://doi.org/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* (Vol. 6394, p. 639406). International Society for Optics and Photonics. https://doi.org/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). Springer, New York, NY. https://doi.org/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, USA: ACM. https://doi.org/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*(Supplement C), 259–272. https://doi.org/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. https://doi.org/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). Springer, Cham. https://doi.org/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. https://doi.org/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). https://doi.org/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). https://doi.org/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*, *17*(2), 300. https://doi.org/10.3390/s17020300

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, Berlin, Heidelberg. https://doi.org/10.1007/11807964_8