E2.1. Documento de necesidades

WP2. User needs and system requirements
Report
Confidential

Version	Date	Description of main changes	Author
1.0	2020/09/28	First draft	UD

Table of Contents

1. Introduction

2. AGROSOFC objectives

3. FUNCTIONAL AND TECHNICAL REQUIREMENTS

This section introduces the methodology followed to specify the technical requirements which consists of the following steps:

- Provide a wide description of the greenhouse:
  - a map of the pilot
  - energy (heat and electricity) demand
  - photos of all the previous items
- Provide a technical description of all the already running monitoring and control system in which AGROSOFC should be integrated

3.1. Description of the pilot

For the development of the pilot activities to be carried out in 2022 and 2023, the initial objective was to develop it in a production greenhouse in the province of Bizkaia, in Spain. The initial idea was to replace the heating system that normally operates in the selected greenhouse, with the technology being developed in the AGRO-SOFC project and evaluate whether this technology is capable of maintaining the necessary temperature conditions inside the greenhouse so that the crop grow properly.

After carrying out different conversations with the producers in charge of the management of the greenhouse selected for the targeted pilot, and after assessing the risks that for the production it can suppose to replace the current heating system by a new system powered by AGRO-SOFC technology, it has been stated that the pilot would be carried out in a small greenhouse specially designed for experimental activities, but with the same technical characteristics as the production greenhouse originally selected for the pilot. It must also be taken into account that the power of the SOFC System was not going to be enough to generate the energy necessary to regulate the temperature of the entire greenhouse, so it would be necessary to operate with two heating systems working in parallel.

According to this situation, Inkoa decided to build a greenhouse tailored to the requirements of the AGRO-SOFC project, with an area of approximately 60 m2, and located next to the production greenhouse that had originally been selected for the pilot. From a climate and irrigation automation point of view, this new greenhouse will operate with the same features than a production greenhouse. The heating system will be regulated according to the same control algorithms that currently operate in the production greenhouse.

To capture the initial data used to simulate the performance of the AGRO-SOFC system, the production greenhouse initially selected for the pilot was used. These data have basically been information about the climatic conditions inside and outside the greenhouse in the last 3 years.

In the following sections, it will be described the production greenhouse from which the information for the simulations has been obtained, as well as the initial design of the small experimental greenhouse where the pilot will be carried out.

3.1.1. Reference greenhouse for data analysis

3.1.1.1. Location and main characteristics

It is a commercial greenhouse located at Munguia (Bizkaia), in the North of Spain. The coordinates are 43,352847, -2,806338 (Google maps) and the total surface of the greenhouse is 2.200 m2 (80m x 28m).
It is a curved roof multi-span greenhouse with a metallic structure and a glass cover. More specifically the greenhouse has five modules of the following dimensions: 5 m in ridge height, 3 m in gutters, 8 m wide and 80 m long. The figures below show the outside view of the greenhouse.

Figure 1- Outside view of the greenhouse.

The greenhouse has a control room of 60m2 (7,5m x 8m) and around 2.140m2 of greenhouse surface for cultivation purposes. The control room is home to the climate control system (Climagro, designed and developed by inkoa) and to the fertigation system, including the tanks for fertilizers and acid, and the fertigation control system (Climagro).
The cultivation area of the greenhouse has no internal divisions, but there is a central corridor, which separates the upper and lower sections of the greenhouse in two sectors (sector 1 of around 1.100 m2 and sector 2 of around 1.100 m2), where fertigation is controlled independently (sections one and two marked in orange in the figure below).

Besides, next to the greenhouse, there is a storage room and a booth with a computer connected to the climate control system, from which the configuration of Climagro can be modified and where the historic data is stored in a SQL database. The figures below depict the aerial view and the simplified blueprint of the greenhouse.

Figure 2- Aerial view of the greenhouse

The greenhouse is used for the cultivation of **tomato** (*Solanum lycopersicum*). More specifically the variety grown in **Jack tomato**, a hybrid of indeterminate growth from which very good results are obtained on the Cantabrian coast. It is a plant with very short internodes, with little foliage and a great aptitude for setting, and it is especially recommended for greenhouse production. It produces large red fruits with a slightly green neck, its skin is smooth and fine and its pulp is juicy and has a good flavour. The cropping calendar of Jack tomato is presented in the table below.

Jan	Feb	Mar	April	May	June	July	Aug	Sep	Oct	Nov	Dec

Plantation											
			Harvest								

Table 1- Jack tomato cropping calendar

More specifically, the greenhouse works with two tomato growing seasons per year:

- Season 1: January/February- early July (transplanting date between January, 10 and February, 20)
- Season 2: Mid July-December (transplanting date around July, 10-15)

Figure 3 - Interior view of the greenhouse (Tomato plants-April 2020).

The tomato is grown in a soil-less system consisting of expanded perlite substrate grow bags with drip irrigation. The expanded perlite substrate bags from PERLINDUSTRIA are of 2,8kg each and they house six plants and count with four emitters each. The greenhouse has 1.150 substrate bags that house around 6.900 tomato plants, while the drip irrigation system counts with 4.600 emitters.

The expanded perlite is a white granular material that is extremely lightweight and has remarkable thermo-acoustic qualities. The countless micro-pores of dry air in the structure makes the material behave as a very efficient natural insulator. The expanded perlite retains the original qualities of the natural mineral; it is chemically neutral, inert to atmospheric agents and the passage of time, non-combustible, non-toxic, insoluble and non-hygroscopic. That makes the perlite as an efficient natural insulation for the hydroponic culture.

Figure 4: Perlite substrate grow bags employed at Munguia's greenhouse.

3.1.1.2. Climate Control

The climate inside the greenhouse is controlled by means of three systems: the heating system (based on fuel boilers), the zenith ventilation and the shading screens.

Heating system-The greenhouse heating system consists of a 750 KW boiler with floor pipes and four air heaters. The boiler is automatically activated and/or deactivated based on the value of air temperature inside the greenhouse as explained in the following section.

Each boiler has a PLC for its automatic activation and/or deactivation based on the value of air temperature inside the greenhouse (measured with an integrated PT100 temperature probe). The temperature set point is usually set at 16°C.

Figure 5 – Main Boilers for the heating System

Figure 6- Heating system- Air heater

Technical characteristics Model Y-200		
Heating power	Max.	750* kW
	Min.	125* kW
Electricity consumption	Switch on	9665 W
	Normal regime	350 W
Control mechanism		PLC (Programmable Logic Controller) with touch screen
Regulation		Integrated probe, external start / stop signals, independent power and analogue control (4-20mA)
Type of regulation		Modulating with 3 powers + external signal
Regulation range by temperature		0-1.100 °C
Maximum safety temperature		Configurable
Electrical power supply		230 VAC / 50-60Hz
Efficiency		99%
** Estimated power using EN Plus A1 quality pellets and an energy yield of 5 kW / kg*		

Zenith ventilation-The greenhouse has roof vents on both sides of the ridge of each nave. The roof vents are opened along the entire length of the greenhouse.

Figure 7: Roof vents opened- longitudinal view.

Figure 8: Greenhouse roof vents.

Shading screens-The greenhouse has zenith and lateral shading screens. The lateral shading screen sections move jointly with the corresponding zenith screen sections, it is not possible to manoeuvre them independently.

Figure 9: Exterior and interior views of the shading screens folded

The zenith ventilation and shading screens of the greenhouse are controlled by Climagro climate control software (version 3.0 from Inkoa). Climagro controls ventilation and shading systems globally for the whole greenhouse (it is prepared to carry out differentiated control by sectors, but nowadays they control the greenhouse as a single unit). Further details on the Climagro control system are provided in following sections.

The greenhouse counts with one relative humidity and ambient temperature sensor and a solar radiation sensor inside the greenhouse and an outside weather station. The weather station measures the following meteorological parameters: wind speed and wind direction, solar radiation, ambient temperature, relative humidity and rainfall (only presence or absence of rain). All these data is directly sent to the Climagrocontrol system, which processes and stores the data in a SQL (Structured Query Language) database and uses such data for controlling the shading and ventilation systems of the greenhouse.

Figure 10: Weather Station and Industrial PC

Figure 11: Electrical Panel – Analog/Digital Convertor Boards

3.1.1.3. Fertigation system

The greenhouse counts with a drip irrigation system. Each perlite substrate bag houses 6 tomato plants and counts with 4 drippers. The normalized water flow of each dripper is 3 L/h. There are 21 drip lines and 1.150 substrate bags, leading to a total number of 4.600 drippers and 6.900 plants in the greenhouse. Pipelines carry water through the entire irrigation system, from the pump through the valves, and onward to the drippers.

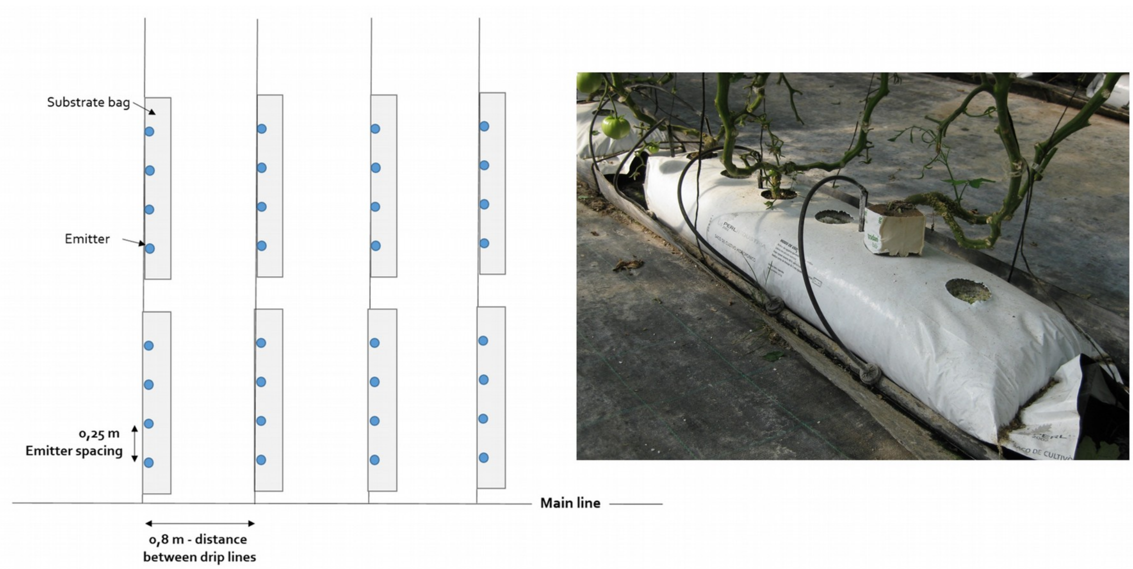


Figure 12: Simplified layout of the drip irrigation system of Munguia's greenhouse and picture of a substrate bag with the four emitters.

The greenhouse fertigation system counts with four fertilizer tanks of 1000 litres, one tank for acid solution (Nitric acid solution) and a mixing tank. The fertigation process is controlled by the Climagro Fertigation Control System *from Inkoa*, to ensure a precise supply of water and fertilizers to tomato according to conductivity and acidity parameters.

Figure 13: Fertigation System and Drainage System

They currently only use 2 fertilizer tanks (Tanks A and B), whose average composition of the fertilizer solutions used is as follows:

- Fertilizer tank A-Composition per 500 litres of water
  - potassium nitrate: 28,5kg
  - calcium nitrate: 30,8 kg
  - microelements: 1,5 kg
- Fertilizer tank B-Composition per 500 litres of water
  - potassium sulfate: 11,20 kg
  - magnesium phosphate: 16,25 kg
  - monopotassium phosphate: 10,20 kg

The Climagro, used for the automatic control of fertigation, has integrated pH and CE sensors to adjust the values of these parameters in the irrigation water. Moreover, Climagro has a solar radiation probe to enable irrigation by accumulated solar radiation. Furthermore, the greenhouse has a demand tray system with a level sensor to be able to correct the irrigation percentage based on the actual drainage. This technology consists of a tray made from metal that contains two crop units (substrate bags). The drainage from the substrate accumulates in a bucket, where there is a water level sensor.

For fertigation purposes, the greenhouse of Munguia is divided in two sectors of around 1.100 m2 each (approximately 40m x 25m), separated by the central corridor. The fertigation scheduling is by time, they normally work with 5 minutes irrigations. Additionally, as a “safety” measure, irrigation is also programmed according to accumulated radiation.

Furthermore, the greenhouse workers have portable pH and CE meters to check the EC and pH values in the tank, in the clean dripper and in the drain tray in order to adjust the duration or frequency of irrigation. If high values of these parameters are detected, an additional irrigation is carried out to wash or the concentration in the irrigation is lowered.

3.2. Description of the current climate control system in the greenhouse

3.2.1. Physical elements

The possible physical elements involved in the climate control of a greenhouse are:

3.2.1.1. Systems

- Window
- Shade screen
- Heating
- Humidifier

- CO_2 Injector
- Cooling System
- Refrigeration
- Analog 3-Way Valve
- Digital 3-Way Valve
- Photoperiodic Lighting
- (Photosynthetic Lighting)
- Air freshener
- Digital Output
- Analog Output

As previous experience of including new types of systems in the Climagro and their practical operation has shown, it has been concluded that what defines each system is more a combination of basic parameters and functions than specific characteristics. In other words, by combining a series of basic options we could define the nature and behaviour of almost any system. Therefore, there will be a single "universal system" configurable through a series of predefined parameters in the form of templates. These templates will physically be xml files. In this way, Inkoa staff will be able to generate new systems or they will be able to adapt to specific particularities of a certain installation by creating variants of the existing templates. This is intended to eliminate the need to compile the program to introduce new functionalities to existing systems or to add new systems.

The most of these details won't be accessible to the end user, who will see only the elements of the installation like windows, screens, etc. without complicating with the internal peculiarities of each system. For the user, the graphical interface will be simpler because he/she won't see more than the parameters he/she can manipulate.

This "universal system" in terms of its particularization can be digital and analog and, in turn, can be divided into three large groups according to the way of regulating the associated outputs:

- **Position control:** with two outputs: windows, screens…

- **Frequency control of an output:** heating, fog…

- **Composite control:** this type of system is new. This would be a combination of the previous two, thus increasing the possibilities of the system while maintaining the same operating philosophy. We would have a system with two frequency controlled outputs. For example, we could define a temperature system with one output associated with a cooling system and the other with a heating system. An output would act in a positive direction with respect to the variable to control and the other in a negative direction. In this way, we would have two "physical systems" operating as a single "virtual system". Some of the possibilities offered by this new type of system are the following:

- Double PID controls could be implemented in this way, depending on whether the output was positive or negative, they would act frequently on one or the other of the outputs. A "dead zone" could be defined to avoid overlapping between the two systems when the measured variable is very close to the set point.

- This type of system can also operate in a more traditional way by establishing a "margin" and a conventional control algorithm. In this last way we would be defining the two set points (set point + margin and set point - margin) but since the control is combined we avoid the user of contradictory or overlapping values.

- The two outputs could also be defined as positive or negative, thus controlling a multi-stage system.

The systems will also have parameters related to the hours of use, number of starts, etc,, to make it possible to monitor the **maintenance** of each system and to enable another module relating to **energy expenditure** considerations which will undoubtedly be of interest to production facilities.

3.2.1.2. Smart systems

With the term "smart systems" we refer to systems with a certain degree of complexity in their operation, such as air conditioning equipment and boilers, which incorporate an electronic control device from the factory.

The direct integration of systems with their own control element is not contemplated in the current version of the Climagro, but it could be an option to take into account in a second phase of development of the program.

Manufacturers of this type of equipment usually offer as an option the possibility of incorporating a Modbus communication module in the controller. The possibility of using a second communication network for the Climagro to communicate with these device could be considered. In this way two things could be achieved:

- Send the corresponding activation instructions directly to the local control device, letting the device be responsible for dealing with the technical particularities of the system.
- Do a supervision of the different internal parameters of the machine. This would be particularly useful in rapidly diagnosing machine breakdowns, because precise data of its operation over time could be sent to the technical assistance personnel of the equipment.

The implementation of this possibility shouldn’t be too complex because Modbus communications are based on accessing to a series of controller memory addresses to read or write. The definition of a "smart system" in Climagro would consist of a table that would relate memory addresses to read or write parameters.

3.2.1.3. Sensors

Sensors for climate control:

- Ambient temperature
- Substrate Temperature
- Ambient Humidity
- Substrate Humidity
- Radiation
- Luminosity
- Wind Speed
- Wind Direction
- CO2
- Rain
- Digital Counter
- Card Counter
- Pressure
- Bucket Demand
- Digital Input
- Analog Input
- Pipe Temperature

The same concept of "universal system" explained above will be applied for the creation of a "universal sensor".
The system will also allow the state of a system to be available for installation control as an additional sensor.
As explained throughout this document, the sensor object, within the control program, will have three groups of properties:

- **Sensor value:** relative to sensor value, signal filtering, etc.

- **Reliability:** a measure of the percentage of "valid" values per time unit.

- **State of alarm:** conditions that determine if the sensor value should be considered an alarm.

3.2.1.4. Links

The links, as in previous versions, define the connection between the operation of a system and the value of the sensors associated with it. In other words, the link contains parameters that indicate to the control system how it should act on the system taking into account the values of the relevant sensors. These parameters are grouped into the following categories:

- **Type of link:** it determines the prevalence of some links with respect to others among those associated with the same system.

- **Type of sensor function:** it determines how the "effective sensor value" is to be calculated based on the values provided by each of the sensors associated with the link.

- **Control algorithm:** on/off, PID, predictive… The possibility of making an indefinite number of sensors part of the link, as explained below, will allow a great flexibility in control strategies. It will be possible, for example, to make temperature controls that take into account not only the internal also the external temperature, to take the average of several sensors, to take into account the historical evolution of certain variables, to enter weather forecast information, etc. This particular section, the control algorithms, will be intended to be particularly flexible in terms of the number of control options and strategies. During the implementation of the application, the different control algorithms that may be included will be studied and the possibility of adding others in the future will be left open.

- **Action in case of alarm / lack of reliability of the sensors:** by default the system activation will reduce to 0%.

Three types of links are defined:

- **Safety links:** they limit the action of the systems following safety criterion regarding environmental parameters.

- **Control links:** they establish how systems should act to bring the value of the indoor environmental variables to the desired values.

In the previous version of Climagro, the links related a system with a sensor to control, evidently, a variable. In the new version each link will relate a system with N sensors to control a variable. We can define how each of these N sensors will influence the control algorithm to determine the activation of the system, in order to control that variable.

In this way, the N sensors that are part of a link can have one of these three functions defined:

- **Primary / secondary sensor:** the link between the N sensors will be fixed to the value of the primary sensor for the calculation of the control algorithm, and any of the secondary sensors may be used as an alternative when the primary is detected to be malfunctioning, as it will be explained below.

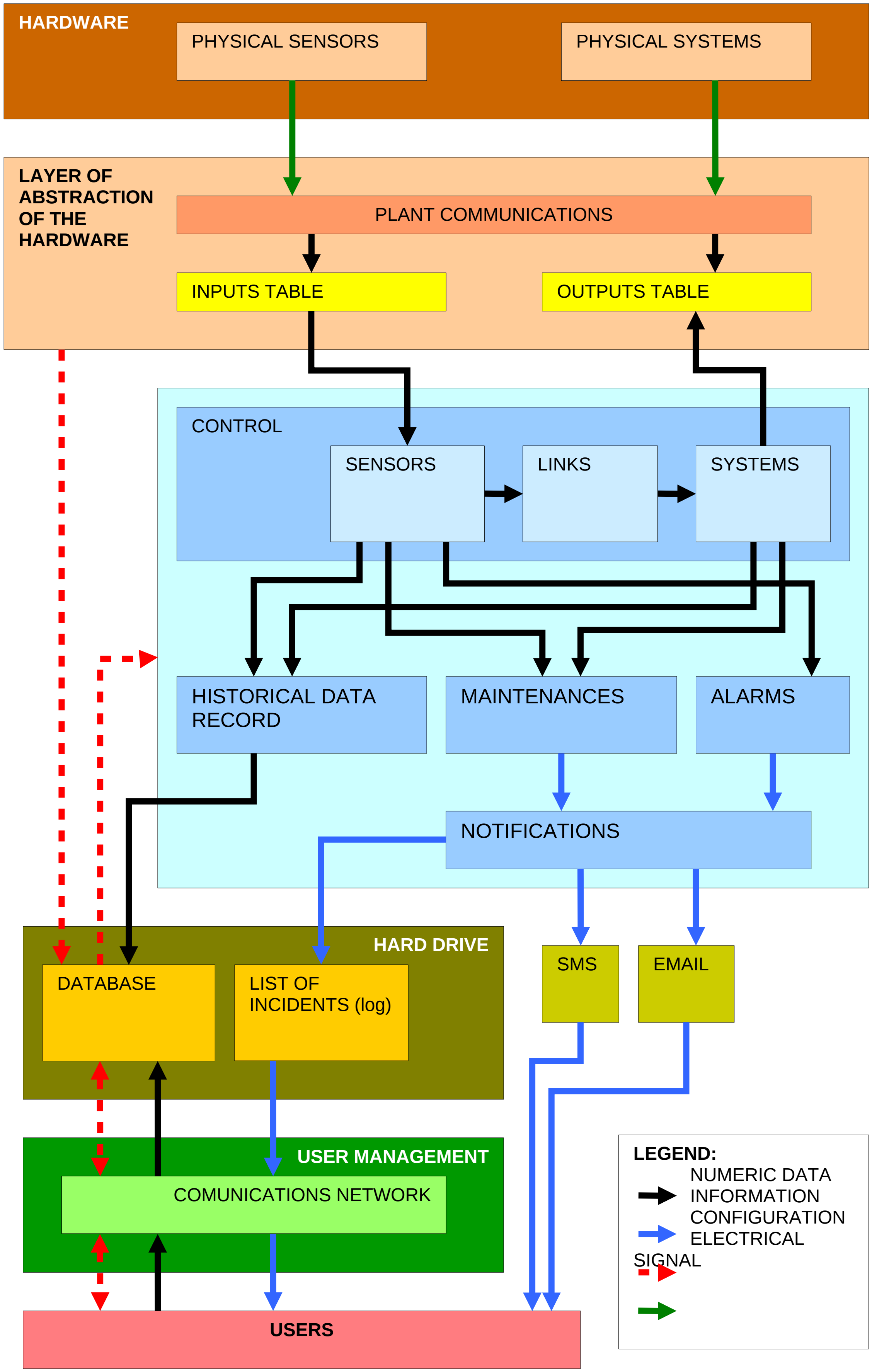
- **Average of values:** the link would take the mean of the value of the associated N sensors as the value of the measured variable for the control algorithm.

- **Special links:** the links that need the value of two sensors of different type to calculate the value of a third environmental variable whose measurement can't be obtained directly. For example DPV, humidity measuring dry bulb temperature and wet bulb temperature. It can also be used in cases where the action of a system is linked to the relation between the measurements of several sensors, rather than to their individual values, which could make their treatment as simple links inappropriate, for example, wind securities applied to windows, which depend on the relation between wind speed and wind direction.

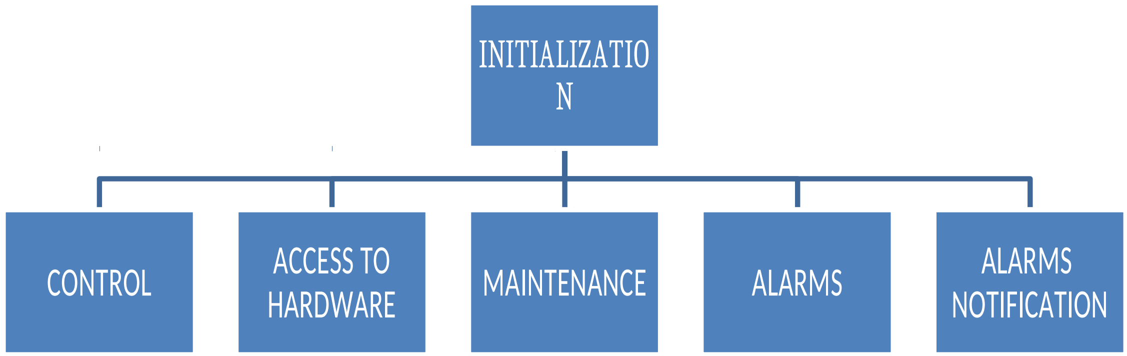
In other words, in the old version of Climagro in each link the value of a single sensor is taken into account. What is being proposed now is to increase the flexibility of the system in the sense that the link can work with the value of alternative sensors if necessary or depending on the combined value of a certain number of sensors.
The link will also keep the parameters relating to the start times of the timetables and the different set points for each of them.

3.2.2. Functionality

The diagram on the next page, without pretending to be a detailed description, attempts to give an overview of the different functional units that will be part of climate control and the interrelations between them.
The hardware abstraction layer communicates the physical elements of the installation with the control logic, translating the electrical signals with which the former are handled into numerical values understandable by the latter. Input and output tables abstract the logical layer from the configuration details of the physical part.
In the control module, the sensor element would filter the data obtained and would verify its validity. This would be the input information for the link element, which have the responsibility of the control algorithm itself. Depending on the activation level obtained, the system element will act on the corresponding outputs.
In the control module, a series of auxiliary sub-processes will monitor the correct operation of the system, generating the appropriate notifications, and making a historical record of all the information related to the operation of the installation for subsequent consultation or management.
A user management layer will provide an interface to access the system state and configure all its operating parameters.


HARDWARE
PHYSICAL SENSORS
PHYSICAL SYSTEMS
LAYER OF ABSTRACTION OF THE HARDWARE
PLANT COMMUNICATIONS
INPUTS TABLE
OUTPUTS TABLE
CONTROL
SENSORS
LINKS
SYSTEMS
HISTORICAL DATA RECORD
MAINTENANCES
ALARMS
NOTIFICATIONS
HARD DRIVE
DATABASE
LIST OF INCIDENTS (log)
SMS
EMAIL
USER MANAGEMENT
COMUNICATIONS NETWORK
USERS
LEGEND:
NUMERIC DATA INFORMATION
CONFIGURATION ELECTRICAL
SIGNAL

The basic tasks that the climate control module will do are as follows:



3.2.2.1. Loading of the installation configuration

When the application starts, the variables necessary to do the control algorithm will be created in memory. The database tables containing the installation definition will be accessed and loaded into memory.
The first data read from the sensors, especially if averages are applied, will not be valid, therefore it is necessary to delay for a few seconds the start of the control loop until the input data can be considered valid.
After this the timers will be started to regulate the running of the installation:

- Control
- Alarms
- Access to Hardware
- Maintenance

3.2.2.2. Reading with self-diagnosis

In each iteration of the control algorithm, each sensor will do a reading of the “Inputs” table loaded by the Hardware Access Layer. The values contained in this table are the results of the A/D converters of the cards. The reading will do a verification of the measurement obtained based on the following sequence:

- Conversion of the table value to user units (°C, HR…) by applying the corresponding calibration parameters (Offset and Span).

- Check if the measurement is within the permissible range for that sensor.

- Check if the measurement has changed with respect to the previous ones, even by a few thousandths of a unit. If the measurement is exactly the same, it may be that the card is freezing and is returning the last value that it calculated.

- Possibility of considering 0,00000 as a wrong value.

The purpose of these checks is to improve the performance of the control in the sense that the validity of the input data, in addition to its value, is taken into account in such a way that the system, in the event of a failure in the sensors, or in the transmission of the value thereof, is able to determine whether it has the possibility to continue operating with alternative input data or has to disable certain systems to avoid the risk of further unintended consequences.

A percentage of erroneous readings shall be kept for subsequent management. An "average" (algorithm to be defined) of the latest correct measurements shall be taken as input for the control algorithm. The sensor will have an associated "operating state" so that the reliability of the returned measurement value can be taken into account at all times. For each link you can define which is the permissible "operating state" of a sensor, this will allow you to decide whether to resort to reading an alternative sensor or to stop the system linked by security, for example.

3.2.2.3. Control

A system can have several links to different sensors.

Control algorithms are methods of the link type. They will use as input data the latest calculated measurement of the sensor or associated sensors, obtained as explained above, the current set point and, depending on the control algorithm concerned, previous sensor values, system activation level or auxiliary values... always checking the operating status of the sensors. The output of this method will be the system activation level. If it could not be calculated because all its sensors are in a non-operating state, it will return a "safe value" that has been previously defined.

Some of the control algorithms that will be included are the following:

- On / off
- On / off with hysteresis
- Proportional Integral Derivative
- Predictive
- Robust
- Fuzzy logic

Once a level of action has been calculated for each link in a system, it must be decided which of them is applied, following the general criterion: the highest of the control limited to the value of the lowest of the security.

The design should be open up to the possibility of incorporating control algorithms that take into account energy efficiency parameters, seasonal variations or historical trends, among other possibilities.

3.2.2.4. Access to Hardware

The reading of the sensors and the writing of the outputs won’t be done directly on the functions that access the hardware, they will be done on the "table of inputs" and on the "table of outputs". These tables will consist of two arrays or matrixes that will function as a hardware abstraction level and will be updated periodically by the hardware access layer.

In the new version of Climagro we are considering the use of sensor networks whose way of communication is different from that of the cards that have been used until now. This forces Climagro to be able to obtain data from different reading protocols for the same installation.

The hardware access layer will be responsible for communicating with the different systems based on the configuration established in each case and will write in memory a specific array to which the control will access for the read/write of data. Thus, regardless of the hardware installed, the access layer will take care of writing the readings made in a known format.
The configuration of the different installed hardware will be done in an application configuration file parameterized by Inkoa when performing the installation. This means that it will be possible, for example, to combine an RS-485 network connected to the COM2 port of a PCI card to RS-485, with a sensor network connected to the USB COM3 port, etc ...
Since this layer has knowledge of the hardware configuration, it will also control errors in communications, not only in total, as is done now, also individually by card. In this way, the diagnosis of communication problems will improve and more precise information will be available for technical assistance.

3.2.2.5. Alarms

The functioning of climate alarms changes somewhat, conceptually, with respect to the previous version of Climagro. The alarm module has here the function of showing an unusual situation in a sensor. That is, the alarm is a state of the sensor. The alarm module will verify if there are the conditions that determine whether a sensor is in an alarm state and it will generate the necessary warnings to the outside. The alarm module won’t directly influence the control other than activating the alarm situation for the corresponding sensors. The control configuration is based on the parameters of the links, therefore, it will be each link which must determine how an alarm situation affects to the associated sensors.
Alarms are defined on the values read by the sensors and/or on their abnormal operating situation. The user will be able to configure alarms for all climate sensors. The following options will be offered:
Regarding the cause of the alarm:

- **Absolute maximum / minimum** value read by the sensor.
- **Maximum / minimum** value **relative** to the current set point. That is, permissible margins with respect to the set point.
- Minimum admissible **reliability** in the sensor.

Regarding the type:

- **Transitional / fixed:** they disappear when the circumstances that cause their activation or require user intervention to deactivate disappear.
- **Delay on activation / deactivation:** they are only activated / deactivated when the abnormal situation persists / disappears for a period of time longer than the delay.

Outputs:

- **Activation of a digital output.** To which an acoustic or light signalling system can be connected.
- **Registration** of activation / deactivation as an anomaly.
- **Sending of SMS message**

- **Sending email**

3.2.2.6. Incident registration

There are some events that are registered when they happen. It is what in the previous version was known as register of anomalies.

- Start and end time of the day
- Start / end of alarms
- File access error
- Errors in the operation of the program
- Completion of maintenance tasks
- Expiry of the date of maintenance without confirmation
- Software exceptions and errors

3.2.2.7. Saving Historical Data

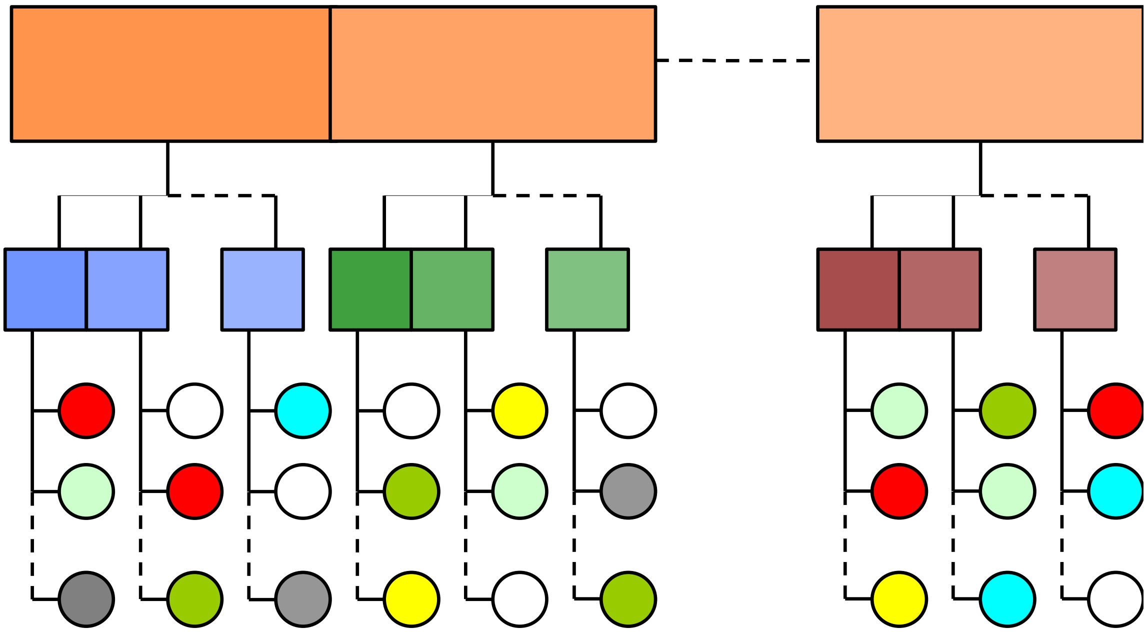
It periodically saves the value of the following parameters for subsequent consultation by the user graphically or through reports and for the diagnosis of eventual breakdowns.

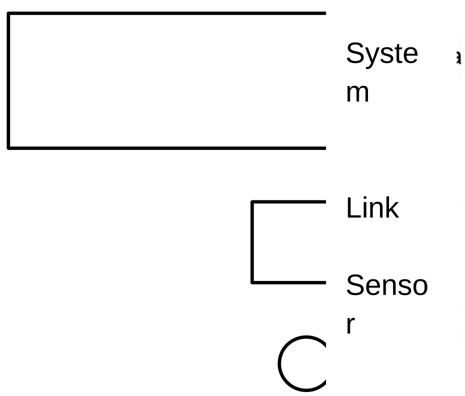
- The value of each sensor
- Operating state of each sensor
- Percentage of incorrect data for each sensor
- Level of actuation of each sensor
- Instructions for the links of each system
- Communications level
- Wrong message rate of each card / device data input
- Operating parameters of "smart systems"

3.2.2.8. Class structure

After being initialized, and for as long as it is running, the program will keep the following object structure in memory:

An array that will act as a container for the instances of objects in the system class. Each system object will have in its properties an array of link objects, which in turn will contain one or more sensors. The same sensor can be part of different links.





The above diagram is intended to be a conceptual representation of the relation between the three central entities of the control system. We will have, on the one hand, an indefinite number of systems. Each of these systems can be associated with an indefinite number of links that combined will determine the system behaviour. Each link can have references to an undetermined number of sensors, from which the necessary data to evaluate the corresponding control algorithms is obtained. The same sensor can, likewise, be providing input data to different links.

3.2.3. Operation of the Heating Control System

Basically the Climagro software doesn’t control the heating, the Climagro controls "whatever has been installed as a heating system", so it can be considered as a positive feature in order to integrate the Agro-Sofc system. This can change enormously from one greenhouse to another. This is the reason for the mechanism of "links" relating systems and sensors. The idea is that each link is a kind of simple rule and that defining the necessary links, the combination of them is adapted to the complexity that a specific installation needs.

To give a more specific explanation: In general, we could say that there are two elements in the typical installation:

- The boiler that produces the hot water.
- The set of pipes and auxiliary elements that distributes the hot water.

Usually the boiler is autonomous and the user starts up it when winter approaches. From that moment the boiler would be responsible for generating hot water according to demand. There is a detail, however, that occasionally the Climagro has been responsible for regulating the temperature of the return water. Basically the water comes out hot from the boiler, at around 70-90 ° C and, after transferring heat along its circuit, it returns to the colder boiler. If the return temperature is below about 55°C, depending on the boiler, it is necessary to drain a small amount of outlet water and bypass it to return so that the water coming back into the boiler is not so cold. Otherwise condensation could be generated that reduces the useful life of the boiler. This is usually done by the boiler control. But sometimes it has been necessary for the Climagro to do it, acting on this bypass based on a water temperature sensor in the return. The boilers can be of very different types: they are usually of natural gas or diesel, but on one occasion of a coal one was installed.

Regarding the distribution of hot water, the simplest case is that it acts on a pump that forces the circulation of hot water through the pipes and/or one or more bypass valves that sectorize the pipes if there are separate areas. A frequency control is usually made on the bypass valve of hot water of a sector based on the difference between the set point temperature and the value that the room temperature sensor is measuring. The algorithm can be on/off with hysteresis or a PID, which will provide a finer control. The pump is configured to start whenever there is demand from a sector, that is, whenever the activation of any valve is higher than 0%.

In a more complex installation, there may be temperature sensors in the heating pipes and valves, instead of being two positions as in the previous case (open-closed), they would be three-way valves. Three-way valves basically allow hot water to be mixed with colder return water by variable opening. This way you can adjust the temperature that you want in the pipe. The three-way valve servomotor can be controlled, depending on the model, with an analog signal (0-10 volts, a voltage proportional to the desired aperture 0-100%) or with two digital signals (opening-closing, adjusting the opening with time given by the opening or closing signal in relation to the full path time). The fact that the same system can have such different types of signals is the reason why the type of output is abstracted in the Climagro and the links generically calculate an activation between 0% and 100%. Then, depending on the output defined by the system, this activation will result in one action or another. It could be the case that a system has different types of pumps.

3.2.3.1. Control Algorithms

The type of algorithm is a property of each link that defines the way in which the sensors associated with the link affect the operation of the system associated with the link. From the programming point of view, each algorithm (PID, basic, on/off, etc.) is an object defined as an instance of a class derived from the class "Algorithm" that serves as an interface to all of them. At each iteration of the climate control program, the link object calls the corresponding function of the control algorithm object set for it.

This function is the called "calculate" which has as input parameters the set point and the value measured by the sensor (or set of sensors) and which as resulting has the activation value for the associated system, a floating point value between 0 and 100. Next, the system object, depending on the number and type of outputs it has defined, will transform the value of that activation into a specific physical action. The overall idea is that the operation is completely modular, that the control algorithm is completely independent of the physical nature of the system that may have associated. In this way, there are simplified the possible corrections that may need to be made in the control algorithms or if any more need to be added

The control algorithm makes use of the following properties:

- **consigAct1:** Set point active 1. For most control algorithms this is the value of the set point at the current moment.

- **consigAct2:** Set point active 2. It is only used in those algorithms that require a second set point value. In other words, in those that define their set point as a margin between two reference values rather than as a single value.

- **margen1:** value, in absolute units of the reference variable, above the value of the current set point. It becomes the maximum activation.

- **margen2:** value, in absolute units of the reference variable, below the value of the current set point. It becomes the minimum activation.

- **parámeteros:** Array of n values. Each of these values is that of a control algorithm parameter. The particular meaning and usefulness of each of them will be determined by the type of algorithm and its position within the array.

- **Signo:** From the point of view that the systems act with respect to the controlled variable, we can classify them into positive systems and negative systems:

- **Positive Systems:** act when the error (as defined) is positive. For example: a heating system with respect to temperature. If we have a set point temperature of 25°C and the temperature measured by the thermometer is 20°C we have an error of 5°C, positive value.
- **Negative Systems:** act when the error is negative. For example, a cooling system with respect to temperature.

Some notes and considerations about the properties described:

- The active set point (the two in fact, although only the first one is used in most algorithms) is a value that is being recalculated in each iteration of the control loop. The user defines the set point(s) as a series of points, values, that must be reached at certain times of the day. The system calculates the instantaneous set point by drawing the straight line between these points. In this way, if the user only defines a single point, the set point will be constant throughout the day. And if the user considers it convenient, he can add as many points as necessary to approximate the evolution of environmental variables to any "curve" without worrying how complicated it is. It is recommended, however, to work with a reasonably low number of set points because otherwise the only thing that is achieved is to increase the load of the computer unnecessarily.

- It may seem that defining an action margin delimited by the active set point 1 and the active set point 2 is equivalent to defining a single set point and two margins, a margin 1 above and a margin 2 below the reference. There is a subtle difference, in the first case, depending on how the user has defined the two set points, the difference between one and the other can vary throughout the day. In the second case the margins are constant values calculated above and below the instantaneous set point.

- **Error = set point - measured value**

- **Activation:** This term refers to the percentage of the power of a system that is used at each moment, with respect to the maximum power. Therefore, activation can take values between 0% and 100%. The main function of the link is to decide what the activation of a system should be. Depending on the type of output associated with the system this will result in one action or another on certain outputs. But that is already the responsibility of the system object. The function of the link is limited to calculate a value between 0 and 100 which we call activation.

Control algorithms:

- PID
- Base
- On/off

Parameters of the algorithms

Control algorithms need a series of parameters that, depending on the case, can be general parameters depending on the type of algorithm or can be adjusted for a specific link. Each algorithm has defined a fixed number of parameters. The definition of these parameters, is collected in two tables of the database:

- cfg_algoritmo: algorithm id and algorithm name

- cfg_algoritmo_parametro: collects the parameters: algorithm id, parameter name and algorithm order number.

These two tables are used by the user interface to know which parameters the user has to demand depending on the algorithm that he/she has selected. These tables are linked to each other but are independent from the rest of the database structure. The values provided by the user are saved in the table cfg_param_link.

The way the control program knows the meaning of each parameter is through their "order". The control program loads, for each link object, an array with the control algorithm parameters.

Base Algorithm

What we are trying to implement here is a slightly more flexible version of the basic control algorithm that was used in the previous version of Climagro.
The previous algorithm used two parameters: the set point and the margin. The operation was very simple. The margin divided by two and applied half over the slogan and half under the slogan, thus obtaining two "extreme" values that corresponded to the extreme activation values (0% and 100%). For example: heating system (positive system), set point = 20°C, margin = 5°C. The operation of the algorithm for this case would be as follows: for temperature values of 17'5°C and lower, the activation would be 100%. For higher values of 22.5°C the activation would be 0%. For intermediate values a linear ratio would be followed, so that for a temperature value equal to the set point the activation will be 50%.
This algorithm has some drawbacks: the margins are symmetrical on both sides of the set point and that the activation corresponding to the set point is fixed and equal to 50%. In the above example of the heating it is clear that the system is not at all symmetrical. The heating actively increases the temperature but, nevertheless, the excess heat provided is only dissipated naturally according to environmental circumstances. On the other hand, a fixed activation of 50% when the temperature is the set point will cause oscillations in most situations.
Therefore, maintaining the same operating scheme, the algorithm has been made slightly more flexible:

- It is possible to define individually the margin above and the margin below the set point:
    - Margin1: margin above the instantaneous set point. Absolute value in units of the variable to be controlled.
    - Margin2: margin below the instantaneous set point. Absolute value in units of the variable to be controlled. The user will supply it as a positive value.

- Parametro[0]: Activation value when the value of the measured variable is the same as the set point. (In the previous Climagro it was always 50%)

- Parametro[1]: minimum value for activation.

- Parametro[2]: maximum value for activation.

Despite being somewhat more configurable, this algorithm still has its limitations. It is maintained because it is practical on many occasions.

		Positive Act. Sist.	Negative Act. Sist.
Margen1		**0%**	**100%**
	Negative error	(1 + e / m1) * X	
Set point		**X%**	**X%**

	Positive error	X + (100-X)(e / m2)	(1 – e / m2) * X
Margen2		100%	0%

E = error
M1 = margen1
M2 = margen2
X =activation when it is in set point, in %

Once the above formulas are applied, the maximum and the minimum are applied.
The parameters of this algorithm are arranged as follows:

Parámetro[0]	X (called k in code)
Parámetro[1]	M1
Parámetro[2]	M2

PID 1

Proportional Integral Derivative Control. It replaces the “advanced control” of the previous version of the Climagro, which used to be a PI control.
In this algorithm the "sign" is that of the link. The parameter "Kc" is assumed to be always positive. If the link is positive, the control algorithm will take the Kc as is, if the link is negative, the algorithm will be initialized by multiplying "Kc" by "-1".
The parameters of this algorithm are organized as follows:

Parámetro[0]	Kc	Gain
Parámetro[1]	Ti	Integral time

Parámetro[2]	Td	Derivative time
Parámetro[3]	b	0 or 1 proportional action options
Parámetro[4]	c	0 or 1 derivative action options
Parámetro[5]	salMax	
Parámetro[6]	SalMin	

The first thing to do is calculate the following auxiliary parameters:

Ki = Kc / Ti

Kd = Kc * Td

The error is calculated differently depending on the term:

Ep = b * sp – pv

Ei = sp – pv

Ed = c * sp – pv

In this way, if b = 0, the set point is not taken into account when calculating the proportional error. This causes that set point changes are not taken into account. To compensate for this modification you may need to increase the value of Kc. The result is assumed to be that it improves the response to disturbances in terms of duration and amplitude of their effect.

If c = 0 the influence of the set point in the derivative error calculation is cancelled. Taking into account that the set point is usually constant for long periods of time, and that the derivative of a constant is zero, this modification would not have much effect on the functioning of the algorithm. The only moment in which there would be a difference would be when there is a sudden change in the set point. This sudden change has undesired effects as it would cause an abnormally high value of the derivative part. What is known as "Differential kick". By not taking the set point into account, this anomaly disappears.

Integral anti-windup: The solution chosen is to add the integral term only if the output of the PID is not saturated, that is, if it has not reached its maximum or minimum value. If the output is saturated, the integral term is added only if its sign is such that it helps to move the output value away from the maximum or minimum value.

Initialization:

```
Ki = Kc / Ti;
Kd = Kc * Td;
```

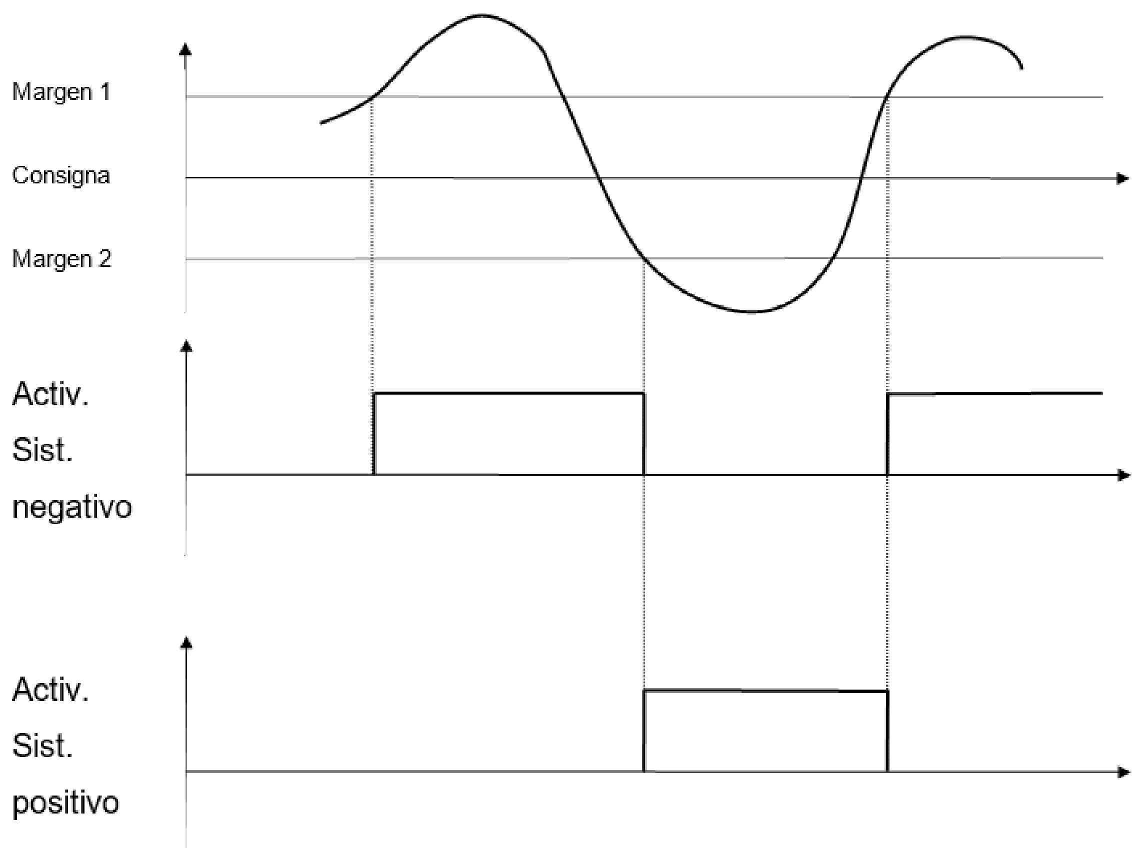
Control loop:

```
Ep = b * sp - pv;
Ei = sp - pv;
Ed = c * sp - pv;

integUltimo = Ei * ((double)Constantes.tBucle);
// ANTIWINDUP
if (sal < salMax && sal > salMin) integ += integUltimo;
else if (sal >= salMax && integUltimo < 0) integ += integUltimo;
else if (sal <= salMin && integUltimo > 0) integ += integUltimo;
```

```
deriv = (Ed - EdAnt) / ((double)Constantes.tBucle);

sal = Kc * Ep + Ki * integ + Kd * deriv;

EdAnt = Ed;

if (sal > salMax) salSat = salMax;
else if (sal < salMin) salSat = salMin;
else salSat = sal;

salAnt = salSat;
return salSat;
```

On / off

This control algorithm is characterized because its output oscillates between the maximum value and the minimum value without ever taking intermediate values.
This is the algorithm to be used with systems such as shade screens. The basic operation consists of alternating between the deployed and retracted position.
Taking into account the criterion of signs established above, it functions as follows:



That is, a "positive" system (see definition), such as a heating, which is programmed to work according to this algorithm, will do the following:

- It will be activated when the temperature drops below a certain value, "Margen 2" and will keep activated until the temperature rises above the "Margen 1" value.

- When the temperature exceeds the "Margin 1" value, the heating is switched off until it drops below "Margin 2".

The parameters of this algorithm are organized as follows:
Parámetro[0] Margen 1 (absolute value, positive number)
Parámetro[1] Margen 2 (absolute value, positive number)
Parámetro[2] maximum, activation value corresponding to “on”
Parámetro[3] minimum, activation value corresponding to “off”

The maximum and minimum values are 100 and 0 by default. However, it may be interesting to give them different values. For example: the chimney effect of the screens can be implemented by defining a security link with a maximum activation of 100 and a minimum activation of 90 related to ambient temperature.

As it is about safety, its effect is to establish a limitation to the activation of the system, in this way we make sure that when the temperature rises above a certain value, the screen will leave a small gap so that the air can circulate.

Digital input

One of the applications of this algorithm can be to alternate the operating state of a system between two values depending on the value of a digital input. Digital inputs can only take two values: zero or one. The set point can only be one, that is 0,5. Thus the input value will always be above or below the reference.

If there is a set point of 0 or 1 the result may not be the desired one because some of the comparisons may be > = or <= and no switching is detected

4. Methodology

4.1. User Requirement analysis and use case definition

The purpose of this section is to perform the requirement elicitation, which comprises a set of activities concerned with the identification and communication of the purpose of the system. These processes are part of the Requirement Engineering procedure. Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a system, and the contexts in which it will be used, among different stakeholders. RE refers to the process of defining, documenting and maintaining requirements of a system. Hence, it is necessary to construct a bridge between user or real world requirements and the capabilities and opportunities afforded by technologies (Figure 1).

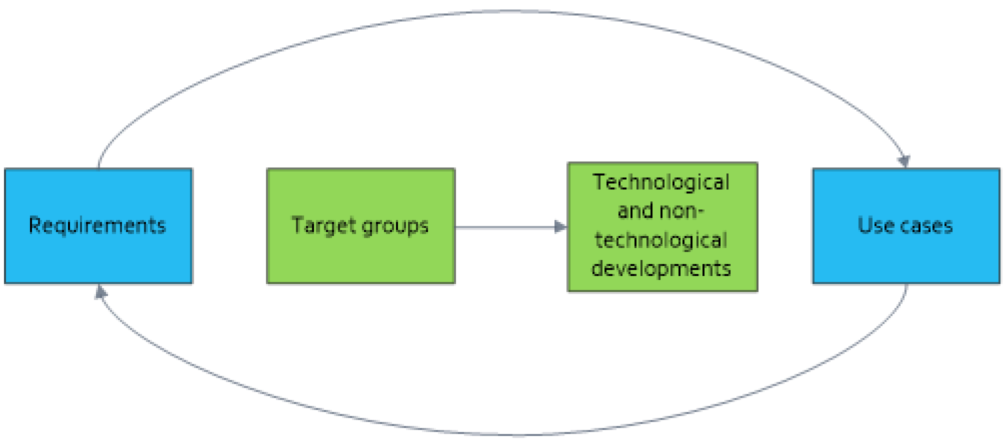


Figure 1. Relation between requirements and use cases

This section begins by outlining the requirement specification methodology used. The goal of this method is to suggest a formal framework to describe the system requirement. Subsequently, the functional and non-functional requirements are listed. Due to the early stage of the project these requirements and the corresponding use cases may suffer from changes that will be updated in the corresponding appendixes.

4.1.1. Requirement analysis

Requirements analysis encompasses those tasks related to determining the needs or conditions to meet for a new or altered product or project, considering the possibly conflicting requirements of the various stakeholders, analysing, documenting, validating and managing software/system requirements. The aim of the requirement analysis is to facilitate the design of the system architecture and the development of the envisaged system.

The AGROSOFC requirement analysis mostly relies on **Volere methodology**. The Volere methodology comprises a general template for presenting the layout and structure of the requirements specification document, as a result of the requirements elicitation and analysis processes. The first edition of the Volere Requirements Specification Template was released in 1995. Since then, several updates have been released that resulted in refinements of the previous ones, based on feedback from an affiliated network of organisations from all over the world that have used the Volere template to discover, organise, and communicate their requirements. The subsequent updates reflect more specific needs imposed by those organisations. The template is distributed under a commercial license and it is copyrighted by The Atlantic Systems Guild Limited. Some older versions are available on the Web.

According to the Volere template of Requirements Specification, the following types of requirements are supported:

- *Functional requirements* which describe which are the functions of the product.
- *Non-functional requirements* which describe the properties that the functions must have, such as performance and usability.

In addition to the requirements types, Volere introduces the following constraints:

- *Project constraints*, which are restrictions on the product e.g. due to the budget or the time available to build the product.
- *Design constraints* impose restrictions on how the product must be designed.

The Volere template also refers to:

- *Project drivers* (in the beginning), which reflect the business-related forces. For example, the purpose of the project is a project driver, as are all of the stakeholders each for different reasons.
- *Project issues* (at the end) including the conditions under which the project will be done. The purpose of this section is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.

AGROSOFC has made used of the template purpose by Waste4Think, an adaptation to the eCOMPASS project project. Therefore, a number of Sections from the Volere template about the description of functional and non-functional specifications were chosen. Table 1 sums up the section used to describe the functional and non-functional requirements.

Table 11. Requirement gathering template

ID	***A unique identifier***
Name	*Title of the requirement.*
Requirement type	*Whether it is a functional or non-functional requirement and in case of non-functional requirements the specific type of requirement according to the Volere notation.*
Description	*A requirement must say exactly what is required.*
Rationale	*A justification of the requirement*
Fit criterion (measurable)	*By measurable we mean is it possible, once the system has been constructed, to verify that this requirement has been met. In other words, this means the tests which must be performed in order to satisfy the requirement.*
Customer satisfaction	*Degree of stakeholder happiness if this requirement is successfully implemented (Scale from 1=uninterested to 5=extremely pleased).*
Customer dissatisfaction	*Degree of stakeholder unhappiness if this requirement is not implemented (Scale from 1=hardly matters to 5=extremely displeased).*
Priority	*The requirement is ranked according to the customer value. (Scale from 1=low priority to 5=highest priority).*
Conflicts	*Any requirements whose implementation is blocked by this one.*
Constraints (attainable)	*An attainable requirement will usually answer the question:* How can the requirement be accomplished?

	Hence, here we provide any constraints / conditions for the requirement to be executed.
Difficulty	*Level of difficulty for requirement implementation (estimation). (Scale from 1=low difficulty to 5=extreme difficulty).*
Actors	*An actor is someone or something outside the system that interacts with it or with one of its components (primary actor). If the actor is interacted by the system or one of its components is a secondary actor.*
Author	*The owner of each requirement that was recorded.*
Version and date	*This field shows the version and date that the requirement was created.*

3.2.2 Use case analysis

Use Case Analysis aims at organizing how external entities interact with the system, emphasizing on gathering the functional requirements of a system. Use cases are an effective technique for capturing, organizing and communicating the functional requirements of a system. A use case typically represents a subset of functionalities of the system towards the achievement of a specific goal, with the key characteristic that these are considered from the user's point of view.

A use case lists a number of possible sequences of interactions between the system and external actors to accomplish a goal, describing a flow of events where the system fulfils one or more user's requirements. An actor represents an external role that interacts with the system (not necessarily a person, it can be any element outside the system, such as an organization, device or even a software package). A primary actor makes use of the system to achieve a specific goal, while a secondary actor provides assistance to the system in order to achieve the primary actor's goal. Table 2 sums up the description of the use cases.

Table 2. Template for the Use Case definition

Use Case ID	<unique identifier in the form UC*x.y*>

Use Case Name	<brief title that expresses the goal of this Use Case>
Description	· Purpose and goal: <intention of the Use Case> · Scenario of use: <detailed statement describing the context, user's profile, why he/she uses the platform, interaction with the system, interaction with actors involved, benefits obtained, etc.>
Primary target group	<the target group who will be initiating this Use Case>
Secondary target group	<other target group who interact with the system/participate in completing the Use Case>
Priority level	<available options: *Essential*, *Secondary, Supportive*>
Reason for assigning this priority	<importance of classifying this Use Case into the specific priority level>
Preconditions	<description of the conditions (if any) that must be fulfilled before the Use Case can be executed>
Trigger	<events that trigger the actions to be executed>
Main flow	<detailed description of the user actions and system responses (in the form of step-by-step action/event list) that will take place during execution of the Use Case under normal, expected conditions> Example: 1. The user sends a request for… … n. The system displays a set of …
System output	<detailed system's functionality as a response to the user's actions>
Alternative flow	<situations that can prevent proper execution of the Use Case,

	referring to the corresponding step number, the condition that must hold true and the alternative actions to be taken> Example: k.a. The system cannot identify the user. k.a.1. The system displays an error message. k.a.2. The flow continues with step m.
Services involved	<AGROSOFC services involved in the implementation of this Use Case>
Devices	<on which device this Use Case will operate properly>
Critical success parameters	<identification of key factors for successful execution of the Use Case>
Constraints	<all constraints to be taken into account, e.g. environmental>
Relevant functional requirements	<IDs of the functional requirements that are used to derive this Use Case>
Legal issues	<comments about Law reference involve in this Use Case (if applicable)>
Ethical issues	< comments about ethical issues involve in this Use Case (if applicable)
Notes and issues	<additional comments about this Use Case or remaining open issues that must be resolved (if applicable)>
Author	<name of the author of this Use Case>
Version	<Use Case version>
Date	<date of last change>

Pilot	<name of the pilot>

3.2.3 Results

The functional and non-functional requirements have been defined for the two expected tools in AGROSOFC project. The ID is made of two fields:

· Tool number Rx, x = {1, 2} where 1 corresponds to the Energy management tool and 2 to the SOFC system tool

· Functional (or non-functional) requirement ID: [N]FRy where y is a multi-index of up to 3 level of depth.

In Annex X the description of the requirements is presented. The colours just represent the level of concretion of the FR. Please note that in order to emphasize the non-functional requirements, they are highlighted in red.

In Annex XX contains the full description of all the Use Cases identified.

5. KPIs definition

Firstly, it has been deepened in the singularities of each software to identify a preliminary list of KPIs by UD. Next, an expert panel composed of the members of the consortium has been consulted for each tool to identify the most relevant KPIs according to the alignment to AGROSOFC project, relation to the pre-defined objectives and milestones and difficulty to calculate or measure them. This process has concluded into a final list of indicators that will be tracked with the AGROSOFC tools over the time.

Finally, a detailed description to ease their implementation in the tools has been made using the template shown in Table X.

Table X.Template for the description of the key performance indicators.

{name of the indicator}	{tier}	{indicator code}

1 GENERAL DESCRIPTION	
VARIABLE	to specify which variable measures the chosen indicator: ● ENVIRONMENTAL ● SOCIAL ● ECONOMIC ● TECHNICAL
STAGE	to specify which stage of the waste management the indicator: ● Energy Generation ● Energy consumption
TYPE	to specify whether it is the DIRECT or INDIRECT indicator
2 CALCULATION METHODOLOGY	
DEFINITION	detailed description of the indicator
STANDARDIZED METHODOLOGY	to indicate whether the indicator comes from a standardized methodology and specify which methodology is
FORMULA	to indicate the formula whereby the indicator is calculated and a detailed description of the necessary variables and data for its calculation
UNIT OF MEASURE	unit of measure of the indicator
CALCULATION PROCEDURE	{to describe necessary information flows for the calculation of the indicator identifying for the primary data ● Information source ● Alternative source ● Measurement location And for the secondary data ● Reference source ● Alternative sources
ACCOUNTING PERIODICITY	To indicate the frequency of calculation of the chosen indicator: daily-monthly-yearly
DATA SOURCE	To indicate the data source

3 PROJECT RELATIONSHIPS					
OBJECTIVES		*MILESTONES*		*SW TOOL*	
4 OBSERVATIONS					
PRELIMINARY INDICATORS	To indicate the required preliminary indicators to calculate the chosen indicator		*SUBSEQUENT INDICATORS*		To specify indicators where the chosen indicator is used
{To specify any NOTES regarding the chosen indicator}					

All KPIs must have a name, an acronym (in English) and a unique indicator code in order to classify them. Annex XX shows the templates fulfilled for all the indicators.

6. .Overview of the architecture

INKOA

7. Validation methodology definition

INKOA