An Autonomous Wireless Sensor/Actuator Network for Precision Irrigation in Greenhouses

Christos Goumopoulos

Department of Information & Communication Systems Engineering, Aegean University, Samos, Greece

Abstract In this work we describe the design of an adaptable decision support system and its integration with a wireless sensor/actuator network (WSAN) to implement zone specific irrigation control in greenhouses via wireless communication. Our research has focused on the provision for proactive applications by deploying sensor networks and connecting sensor data with actuators through an ontology-based decision-making layer. The system developed provides for real-time monitoring and control of both agricultural inputs and outputs (irrigation control). A simple rule editor is also provided through a graphical user interface for the domain-expert to specify the knowledge base.

Keywords wireless sensor actuator network, ontology, decision support, precision irrigation, distributed system, rule-based system

1 Introduction

Thanks to developments in the field of wireless sensor networks (WSNs) as well as miniaturization of the sensor boards many scientific fields can benefit by exploiting WSNs as an effective tool for application research. WSNs are becoming integral to a number of applications, including environmental monitoring [9], building energy management [12] and multi-player games [6]. Agriculture has also become an important application field of WSNs [19]. For example, The LOFAR [1] and PLANTS [11] projects applied WSNs in the precision agriculture domain. Through the use of WSNs fine grained measurements can be programmed at fixed time intervals [2] or based on an event driven model [5]. Also they can be programmed in such ways in order to follow specific routing protocols [13].

At present the information gathered by sensor networks deployed in a field are mainly used to monitoring and reporting on the status of the crops [3, 14, 16]. However, agricultural environments make a good candidate for using proactive-computing approaches for applications requiring a faster than human response time or that requires precise, time-consuming optimization. For example, irrigation is a major issue in many farms. An ideal proactive system would optimize water needs in different areas of the farm with available water, particularly because water is a limited, shared resource. Being able to water plants more selectively and precisely on the basis of individual plant needs and available water, would save water.

In this work we describe the design of an adaptable decision support system and its integration with a wireless sensor/actuator network (WSAN) to implement zone specific irrigation control in greenhouses via wireless communication. Our research has focused on the provision for proactive applications by deploying sensor networks and connecting sensor data with actuators through an ontology-based decision-making layer. The system developed provides for real-time monitoring and control of both agricultural inputs and outputs (irrigation control). A simple rule editor is also provided through a graphical user interface (GUI) for the domain-expert to specify the knowledge base.

In Section 2 we discuss the architecture and design of our system. A layered modular architecture of the system is proposed to enable system flexibility and extensibility. Section 3 gives details of the wireless sensor/actuator network platform developed. Section 4 discusses the ontology-based decision making layer. An illustrative example application of our developed system in the precision agriculture domain is given in section 5. In the next section we discuss the lessons learnt from this effort and finally we give the conclusions.

2 Architecture and design

The overall goal of the research and development effort described in this work was to devise an integrated system in the form of a plant growth monitoring and control system comprising a distributed network of sensors for sensing the plant growth activity and environmental conditions, and plant growth control actuators, an ontology of plant lifecycle domain knowledge and decision support mechanisms that associate sensor data with actuators.

In a high level view the process performed by our system can be seen as a plant/environmental context management process. We model this process as a measurement-translation-reasoning-actuation control cycle (Fig. 1). A mechanism for low-level context acquisition, which reads plant/environmental signals from sensors, starts this cycle. This information is probably not in a format that can be used by the system in order to make decisions or reach a conclusion. In a second phase the signals are interpreted and high-level context information is derived. For example, temperature and soil moisture sensors return an analogue signal (voltage value) which must be then converted, after a calibration phase, to a digital valuable knowledge. This phase is usually performed within the sensor nodes, optimizing both network throughput and battery life by avoiding unnecessary send/receive messages. Aggregation of context is also possible meaning that semantically richer information may be derived based on the fusion of several measurements that come from different sensors. To determine water stress, for example, requires monitoring of plant's leaf temperature, ambient temperature and soil moisture content. The aggregation of context is an operation that is performed at higher levels of the system, usually at the coordinator node.

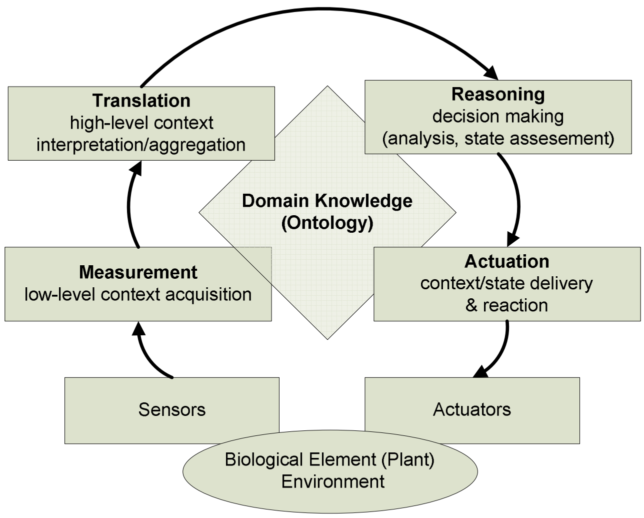

Fig. 1. Measurement-translation-reasoning-actuation control cycle

Having acquired the necessary context it is possible to assess the state of the plant and decide appropriate response activation. This state assessment will be based on a set of rules, which is obtained from plant science research through a manual and time consuming process. The raw (sensor) and high (fused) level data, their interpretation and the decision-making rules are encoded in an ontology. The reaction may be as simple as turn on a light or send a message to the user or a composite one such as request watering to the pot in case of drought stress or as spraying mist in case of heat stress. Such a decision may be based on local context or may require context from external sources as well, e.g., weather station supporting prediction of plant disease spreading.

Fig. 2 illustrates the system architecture in a conceptual view. The wireless sensor/actuator network (WSAN) is implemented as a grid of nodes that communicate and exchange data using radio links. The core technology of a WSAN node is a commodity mote (a battery supplied board with a microcontroller, a memory unit for data storage and a transceiver for communications) connected to a variable number of sensors to provide the necessary monitoring functionality. Furthermore, it is possible to connect actuator units to activate devices associated with agricultural inputs (e.g., irrigation). The network can be configured in a master-repeater-worker operation pattern. The network is synchronized by a single gateway node (master) which collects all sensor measurements before forwarding them to populate a MySQL relational database managed by the coordinator node. Repeaters may be defined to extend the master sync-signal over multiple hops, if necessary.

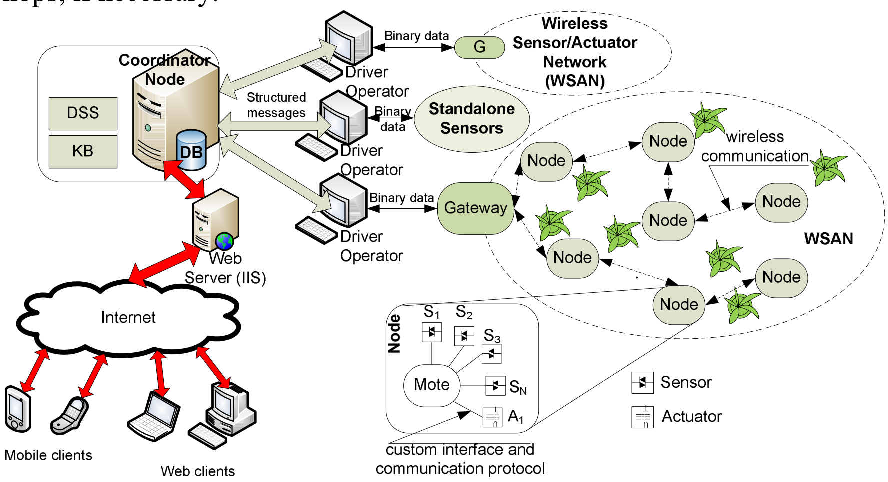

Fig. 2. High-level system architecture

The gateway of the WSAN is connected to an autonomous component which is called the river Operator (O). In the DO a specific driver is designed and implemented for each sensor subsystem (WSAN or standalone sensors) implementing the communication protocol with the hardware. It is also responsible for wrapping binary data coming from the WSAN or low-level requests to the motes into a structured stream of information, including data time stamping, that is transferred between the WSAN and the oordinator Node (N). The functionality of the CN is based on the following core software components: ecision Support System (SS) and nowledge Base (B). The DSS has the responsibility to take all sorts of decisions according to the knowledge acquired with the analysis of the stored information. The KB includes all the knowledge that supports the decision-making process in the form of an ontology.

The high-level system architecture envisages the use of standalone devices that will communicate directly to a DO making use of a specific communication protocol. The standalone devices consist of highly complex sensors such as an infrared imaging system. Such sensors cannot be connected directly to the WSAN due to their complexity, power consumption, cost and data size that have to be transferred to the CN.

Finally, our architecture allows through a web based interface the user to monitor remotely the collected data and request graphs on demand, or to change certain variables of the network (e.g., sampling rate) mainly for performance purposes.

3 Wireless network hardware platform

Developing complete sensor systems for environmental monitoring is currently a complex and expensive endeavor. There is a trend towards leveraging commodity technologies to enable the construction of cost effective sensor platforms, enabled by recent growth in wireless communications and microprocessor controlled consumer devices. This leads to small, low cost, limited functionality devices working in a cooperative networked mode, to form a single system.

Our WSAN was designed using Crossbow's MICAz wireless sensor motes (MPR2400), which are equipped with the Atmel ATmega128L processor running at 8MHz, 2.4 GHz Chipcon CC2420 IEEE 802.15.4 compliant ZigBee radio frequency transceiver, 128KB program memory, 512KB measurement flash, and 4KB EEPROM [7]. A data acquisition board (Crossbow's MDA300CA), was used to process the analog outputs from the sensors and to control actuators. The board has a 12-bit analog-digital converter (ADC) with an 8-channel multiplexer, and a sampling rate of 50 kHz. On this board, seven channels (A0-A6) are single-ended analog channels. The input range for the analog signals is 0-2.5V. Four additional channels are differential analog channels. Three excitation channels and two LED channels occupy the rest slots of the analog channels. For input signals of higher than 2.5V, a voltage divider is necessary to scale down the voltage levels. The result of ADC can be converted to voltage with the formula: `Voltage = 2.5 × ADC-READING / 4096`

Finally, the MIB520CA base station module provides USB connectivity to the central system and MICAz motes for communication and in-system programming. Fig. 3 gives an overview of the hardware platform employed. MICAz and MDA300CA or MIB520CA are connected respectively through the 51 pin connector.

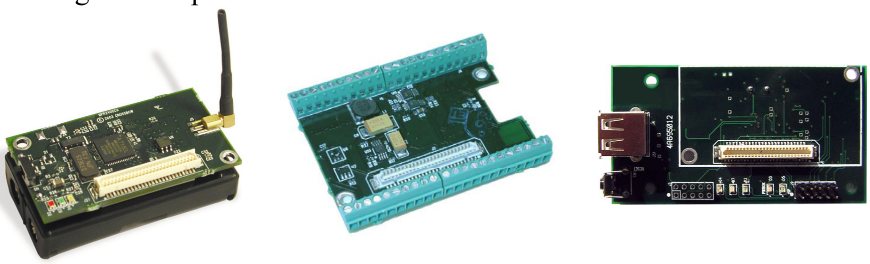

Fig. 3. The hardware platform includes a Mote Processor Radio platform (MPR2400), a data acquisition board (MDA300CA) and a Mote Interface Board (MIB520CA) for network base stations and programming interfaces.

On the software side, the microcontroller runs TinyOS [17], an optimised operating system that allows fast configuration of the sensor nodes implementing communication with active messages [8]. For the implementation of the applications which run on the motes we used the nesC programming language [10].

3.1 Sensor interface

For the greenhouse application domain, the sensor interface portion developed allows soil moisture probes, humidity and temperature sensors to be interrogated by the controlling software running on the mote's microcontroller. Table 1 summarizes the principal features of the sensors that have been interfaced to the mote.

Table 1 Summary of principal features of sensors interfaced to the mote

Sensor	ECHO EC-10	HumiRel HM1500	RS151-237
Measured data	Volumetric Water	Relative	Leaf, air temperature

	Content (VWC)	humidity	
Range	0 to saturated VWC	1 to 99 %RH	-50 to 150 °C
Resolution	0.1 %VWC	+25 mV/%RH	0.1 °C
Accuracy	±1 to ±3% VWC	±3 %RH	± 0.2 °C
Response time	10 ms	10 sec	10 sec
Supply VDC	3.3 VDC	3.3 VDC	5 VDC
Output	450-1550 mV	1325-3555 mV	670.1-185.97 ResΩ
URL	www.decagon.com	www.digikey.com	www.rs-components.com

The requirements for a contact temperature sensor are that the sensor has a large sensitivity to small changes in temperature (≤0.5 °C), low thermal mass, high thermal conduction, low cost and is small enough to place on the plant leaf to measure its temperature. A thermistor is a device that fulfils these criteria and is readily available. Thermistors are made of a piece of semiconducting material that changes its resistance with changing temperature. These can be of two types, NTC or PTC (negative thermal coefficient or positive thermal coefficient). The resistance of these devices also varies. A NTC device with a nominal resistance of 10kW at 25 ˚C was chosen. The analogue to digital conversion on the microcontroller is only capable of reading and digitising a voltage, so due to the changing resistance with temperature characteristics of the thermistor, a voltage divider was used to convert the resistance change into a voltage change. Fig. 4 shows the relationship between temperature and measured voltage with a load resistor that is about half of the nominal greenhouse-temperature resistance.

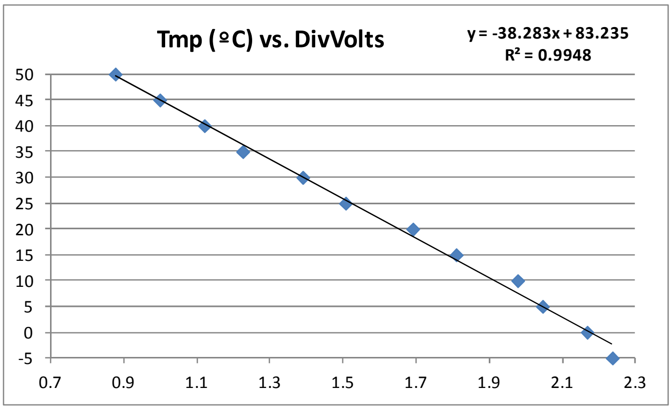

Fig. 4. Flattened thermistor response in divider network

For the humidity sensor interfaced to the MDA300CA we use the conversion formula that is provided by the manufacturer:

```
%H=(int)((-3.9559E-6*adc^2 + 6.1797E-2*adc - 67.681)+0.5)
```

A range of commercial soil moisture sensors are available on the market, but some of the criteria that were required for our application were that the sensor be of small size to fit a 1 L pot (approx. 10 cm length) and have a fast response time. The ECH_2O EC-10 sensor, supplied by Decagon was eventually chosen. EC-10 adaptor cable for connection contains three wires, which are connected to excitation E3.3, single ended analog channel and GND respectively in MDA300CA board. The sensor is 10 cm long and uses capacitance to measure the dielectric permittivity of the surrounding medium. The volume of water in the total volume of soil most heavily influences the dielectric permittivity of the soil due to the relatively higher dielectric of water (80) compared to other constituents of the soil (mineral soil 4; organic matter 4; air 1). This allows the association of the changing permittivity and the soil water content. The sensor gives out a mV signal that is proportional to the water content. In order to get accurate measurements we performed calibration of the sensor in the peat substrate that was used in our application following the process described in [4]. A series of peat/water mixtures were used to obtain data on the sensor output. Sensor readings were obtained for different sensors and repeated 3 times. Fig. 5 shows the calibration curve that was obtained on averaging values from the above experiments.

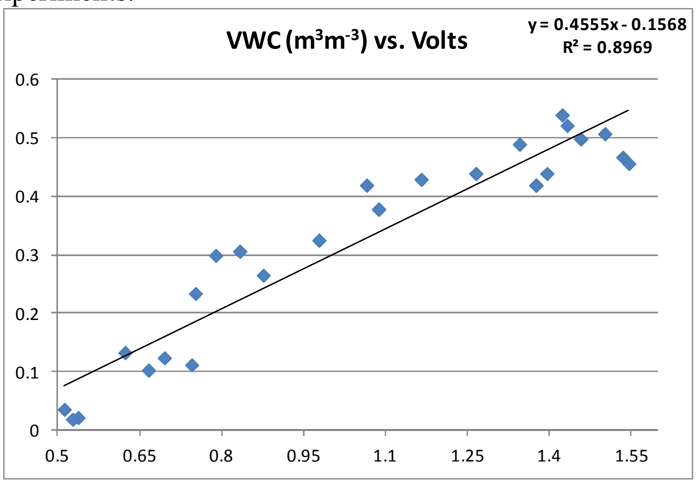

Fig. 5. Plot of sensor output for 3.3V dc supply for calibration of the EC-10 sensor

Fig. 6 summarizes the connections of the sensors discussed above to the MDA300CA module.

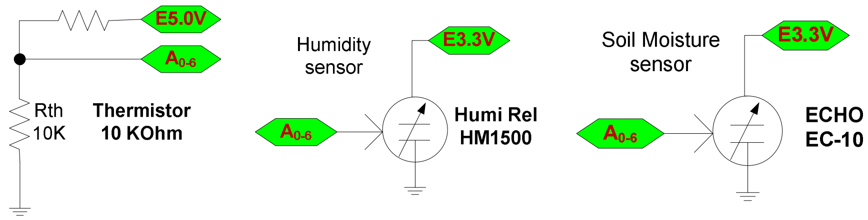

Fig. 6. Greenhouse sensors connections to MDA300CA schematic

3.2 Actuator interface

The MDA300CA module has two relays that can be used for the activation of various devices. In our case it can handle toggling of the irrigation distribution system. In principle, the mote's controlling software, via a transistor switch, activates a relay that activates the device. In the following we describe the irrigation system interfacing.

Fig. 7 shows the irrigation distribution unit. The solenoid operated water valves turn on the plant watering system which supplies four distinct zones of plants. This allows watering only the zone that starts drying. A fifth water valve provides humidity control. When this valve is open, it delivers water to a central spraying system that can be mounted in the apex of the greenhouse to maintain a good level of humidity. The MDA300CA module controls a switch. When this switch is closed it completes the relay circuit causing the electromagnet to attract the corresponding switch to the on position causing the solenoid to open.

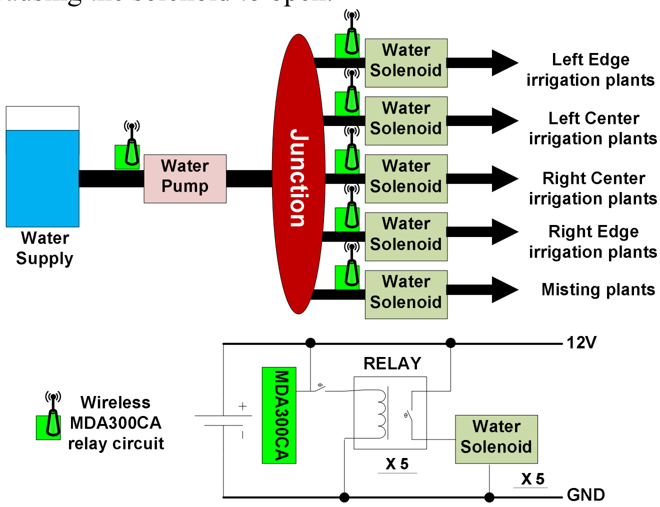

Fig. 7. Schematic layout of the irrigation distribution unit

To control the distribution of the water an application level component controlled by the coordinator node needs to be installed. In order for this system to work the module has to take commands from the master mote in order to know what section to irrigate and for how long. The operation of the irrigation is as follows. The master sends a command via its radio link to the worker mote controlling the actuator. This command is sent to the module in the irrigation distribution unit that converts the signal to a voltage level that the microcontroller can read. Once the microcontroller receives a valid command it will perform a task to turn on a pump and open a particular solenoid in order to irrigate a specific section of the crop layout.

3.3 Message structure

Communication with the WSAN is based on active messages. The message structure is partly predefined by the TinyOS. The packet size includes 7 bytes of generic active message fields and a maximum of 29 bytes for the payload. The payload in our case uses 22 bytes formatted according to the type of the message. The communication protocol implements three message types: request, reply and event-driven data reporting. These message types allow implementing the two general data collection paradigms, commonly known as push and pull. Part of the payload structure of the message is shown in Fig. 8 (we omit the standard fields of TinyOS active messages). The sequence number field is used to ensure that packets will not arrive at the gateway out of order. The source and destination addresses define the message originator and recipient respectively. Address 0xFF is used for the gateway and the range 0x00 to 0xFE is used for the workers. The action type field specifies the action to be performed (e.g., taking measurements, controlling pumps, etc.).

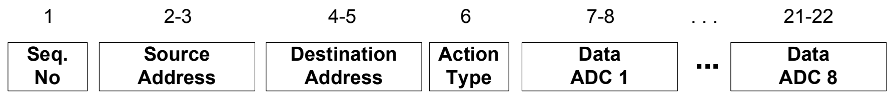

Fig. 8. Part of the message structure

4 Ontology-based decision support

According to Uschold and Gruninger, an ontology is a tool that can conceptualise a world view by capturing general knowledge and providing basic notions and concepts for basic terms [18]. The ontology developed for our system sets up a concept framework on how the knowledge about sensors, actuators and systems available on one hand and the biological studies about plant stressing and sensing mechanisms and consequent plant behaviour on the other hand can be formalized in order to make plants an active part of the resource management process. The decision-making process based on the sensing of plants is also structured for the selected set of sensors and actuators and the correlated biological information allowing interpreting the plant behaviour.

The knowledge that is required by the Decision Support System can be divided into various categories:

Knowledge regarding the plant itself. In this category knowledge such as the name and the species of the plant is described. Additionally this category contains knowledge about the growth and the development stages of plants.

Knowledge regarding plant parameters being monitored by sensors. This category contains information about the available sensors that can monitor the plant parameters as well as relative knowledge like the range of values, the threshold values and the interpretation of the aforementioned values.

Knowledge regarding the state of plants. This category contains information relevant to the plant stressing and sensing mechanisms and the signals that plants perceive and send to the environment. Specifically, the possible states of a plant implied by its parameters and monitored by sensors are part of this knowledge. For example, the representation of stresses, like the water stress, diseases and symptoms belong to this category of knowledge.

Knowledge regarding environmental parameters. The knowledge about the environmental parameters that we can measure and monitor is essential in order to define the state of a plant. For example, parameters like the temperature, the humidity, the carbon dioxide, the light and the soil moisture play a major role. The description of these parameters, their range and threshold values are also represented.

Knowledge regarding sensors/actuators. The sensors and the actuators play a crucial role in precision agriculture resource management. In particular the use of sensors requires a description that specifies their type, the parameter they measure, the range of their values as well as their sensitivity and accuracy.

Rules for decision making. This category refers to the knowledge that supports the decision-making process. This knowledge is represented as a set of rules which are used for various decisions. Firstly, there is a need for a set of rules that will take into account both plant and environmental parameters and the description of a plant in order to diagnose a plant's state. Secondly, the decision-making is based on a plant's state, its description and user defined policies and determines the possible actions of the system, like the request for a resource. The decision-making process will be based on a set of rules in operational representation forms, which will be applied on existent knowledge and allow the use of the ontology for reasoning providing inferential and validation mechanisms. The reasoning is based on the definition of the ontology, by using first-order predicate calculus. The user can define or update existing rules using a front-end tool and expressing the rules with simple if-then-else logic.

Some of the basic concepts represented in the ontology are the following: *Sensor*, *Actuator*, *Parameter*, *EvironmentalParameter*, *PlantParameter*, *PlantState* and *Rule*. In the ontology these concepts are represented as different classes, which have a number of properties. Fig. 9 and Fig. 10 illustrate, for example, subsumption relationships for the concepts *Parameter* and *PlantState* respectively. The specification of the ontology was performed using the Protégé ontology development tool (http://protege.stanford.edu/) based on the OWL standard language.

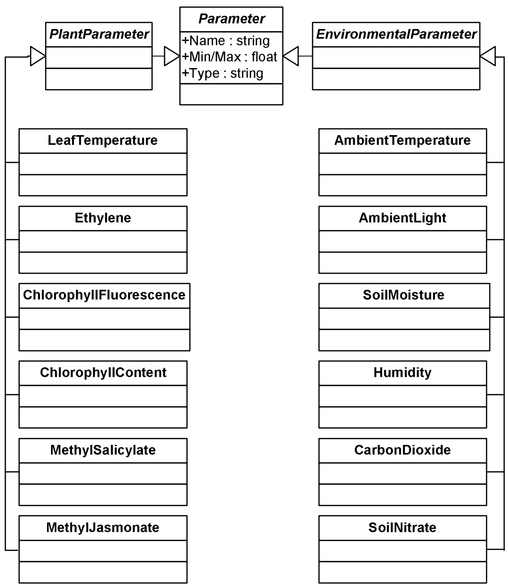

Fig. 9. Subclasses of concept *Parameter*

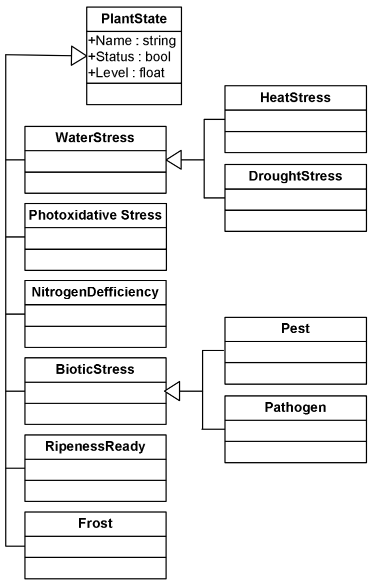

Fig. 10. Subclasses of concept *PlantState*

The outline of the DSS architecture is shown in Fig. 11. The *Process Manager* (*PM*) is the coordinator module of the system and the main function of this module is to monitor and execute the reaction rules defined by the supported applications. These rules define how and when the infrastructure should react to changes in the environment. The *Hardware State Manager* (*HSM*) maintains a repository of the hardware environment (sensors/actuators) reflecting at each particular moment the state of the hardware. The *Communication Module* (*CM*) is responsible for handling the communication with the different device drivers which implement the communication protocol with the WSN or the standalone sensor device.

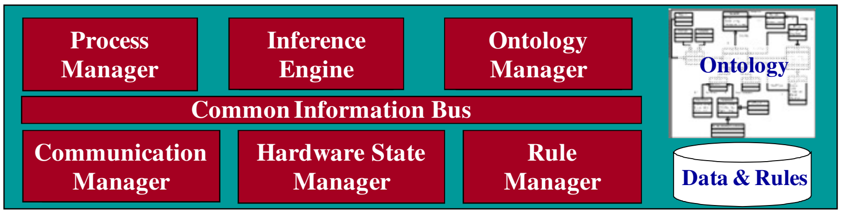

Fig. 11. DSS Architecture

The *Ontology Manager (OM)* module has been defined for the manipulation of the knowledge represented into the ontology and to provide the other modules of the system with parts of this knowledge with a level of abstraction. This means that only the OM needs to understand Ontology and be able to use it; all other system modules can query the OM (through the PM) for the information that they need without any knowledge about the ontology language and its structure. Therefore any changes that may be done to the Ontology affect only the OM and the rest of the system is isolated from them. The OM provides methods that query ontology for the definition of specific concepts, and for the existing instances of specific concepts, like the environmental parameters.

The *Rule Manager (RM)* is the mechanism that manages the rule base of the system and its basic functionality is to provide to the other modules the rules that define an application's logical operation. The basic operations of the RM are to query about the rules of an application and to update them. For the initialization of the decision-making process apart from the rules the initial facts are necessary which represent low level environmental/plant context sensor measurements or inferred plant states. In this respect, the RM is also responsible for the creation of the initial facts of a specific application. For example, an initial fact is the definition of the existence of the plant type with its specific parameters, states and possible actuations that participate in its rules. In order to create this initial fact the RM needs to know knowledge that is stored in the ontology. For that, it queries the OM through the PM for any information that it needs, like what are the parameters, the states and the sensors/actuators.

The *Inference Engine (IE)* is the module of the DSS architecture that supports the decision-making process. This module exploits the Jess (Java Expert System Shell) rule engine (http://herzberg.ca.sandia.gov/jess/). The execution of this module is started based on the initial facts (defined by the RM from knowledge emerged from the ontology through the OM) and the rules stored in the rule base. The IE module is informed for all the changes of parameters values from sensors measurements through the HSM. When the IE is informed for such a change it runs all its rules. When a rule is activated the IE informs for the activation of this rule and for the knowledge that is inferred the PM, that is responsible to transfer this knowledge to any module that needs it.

Regarding the *Rule Base* currently the rules are stored in CLIPS format and the concepts that appear are emerged from the ontology. This is an approach of building rules on top of ontologies. Fig. 12 shows the design of the 'Heat Stress' calculation rule for the irrigation application using a simple rule editing tool targeted for the domain expert that provides a visual interface based on a node connection model.

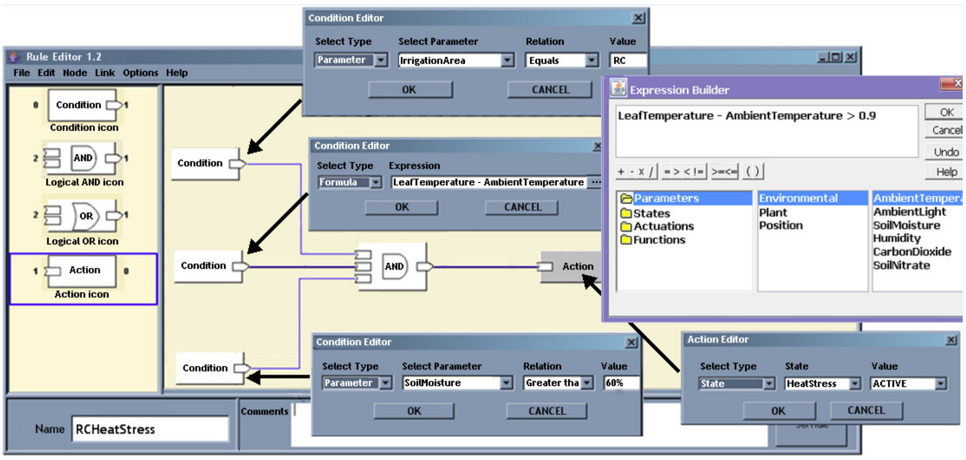

Fig. 12. Editing the 'Heat Stress' rule for the RC area of the irrigation application

The rule consists of three conditions combined with a logical AND node. The first condition checks the applicability of a specific area (RC for Right Center) of the field layout for which we need to evaluate the heat stress state. This is an artificial condition that helps resolving the fact identifiers in the generated CLIPS code. The second condition checks whether the difference between plant leaf and environmental temperature, in that area, is over 0.9° C. An expression builder facilitates the definition of the condition relying on concepts stored in the ontology. The third condition checks whether the average soil moisture in the specific area is over 60%. The rule, as designed, states that when all conditions are met then the heat stress state of the RC area must be set to active (HeatStress ACTIVE).

In Fig. 13 we give the generated CLIPS code for the `RCHeatStress` rule defined with the rule editor. Actually, for each user-defined rule the generated code includes both the 'Activated' part (`RCHeatStressA` shown in Fig. 13) and the 'Deactivated' part (`RCHeatStressD` implied) as the complementary rule to ensure data integrity. The namespace of the facts depends on the irrigation area (e.g., RC) and the resolution is performed automatically based on the `IrrigationArea` parameter selected in the user interface (Fig. 12).

```
(defrule RCHeatStressA
?ePlant<-(ePlant (name ?name) (RCAmbientTemperature ?RCAmbientTempe-
rature) (RCLeafTemperature ?RCLeafTemperature) (RCSoilMoisture
?RCSoilMoisture) (RCDroughtStress ?RCDroughtStress) (RCHeatStress
?RCHeatStress) (RCNeedIrrigation ?RCNeedIrrigation) ... (RCHeatStress
FALSE) )
( test (and (> (- ?RCLeafTemperature  ?RCAmbientTemperature) 0.9)
(> ?RCSoilMoisture 0.6)))
=>
(retract ?ePlant)
(assert (ePlant (name ?name) (RCAmbientTemperature ?RCAmbientTempera-
ture) (RCLeafTemperature ?RCLeafTemperature) (RCSoilMoisture ?RCSoil-
Moisture) (RCDroughtStress ?RCDroughtStress) (RCHeatStress ?RCHeat-
Stress) (RCNeedIrrigation ?RCNeedIrrigation) ... (RCHeatStress TRUE)
))
(call ePlantOS.inference.Entity fireInferenceEngineEvent "RCHeat-
Stress=TRUE")
)
```

Fig. 13. CLIPS code for the Activated part of the RCHeatStress rule

Using a rule editor for defining application business rules emphasizes system flexibility and run-time adaptability. In that sense, our system architecture can be regarded as a reflective architecture that can be adapted dynamically to new requirements. The decision-making rules can be configured by domain experts external to the execution of the system. End-users may change the rules without writing new code. Therefore, the power to customize the system is placed in the hands of those who have the knowledge to do it effectively.

5 System Deployment

The application described in this section is composed of a strawberry plant (Fragaria ananassa) where the plant is controlling irrigation. Irrigation is applied according to the specific requirements of the plants in different parts of the crop array, thus illustrating the precision delivery of agricultural inputs. Fig. 14 shows snapshots of the experimental setup.

The plant/environmental parameters explored for the application development are: Plants' leaf Temperature (PT), Ambient Temperature (AT), and Soil Moisture (SM). PT was chosen as it indicates drought and heat stress [15]. Since we cannot measure the stress itself, we measure indicators of stress, i.e., PT compared to AT and substrate moisture (SM). Comparing the plant and ambient temperature indicates a plant's response to its environment. Coupling these with a SM reading allows determination of whether the plant is heat stressed or drought stressed. Therefore, if the plant has adequate water (determined by the soil moisture probe) but the plant temperature is high this means that it is heat stressed and requires misting to cool it. However, if the temperature is high and the moisture content low, then pot irrigation is required.

Fig. 14. Experimental setup (left); attaching a thermistor to a strawberry (right)

The field setup consists of an array of 96 plants placed in a greenhouse, arranged in an array of 12 by 8 (Fig. 15). The setup consists of 4 different zones: Left-Edge (LE), Right-Edge (RE), Left-Center (LC), Right-Center (RC) and also there is the zone specified for misting which overlaps with the RC zone. The setup integrates the thermistors and the soil moisture probes into one system that can irrigate when required and also determine when to stop the irrigation. Each zone can be controlled using individual solenoids. Misting can be applied to the grey area only for this setup.

A total of 12 Crossbow motes are required to instrument the field: 8 modules are used for connecting the various sensors, each one 'supervising' the sensors in the neighbourhood of an array of 3 by 4 plants; 1 module is sensorless and is used as a gateway with the coordinator node; and 3 modules are used for controlling the irrigation system. The nodes are housed tightly in IP-67 rated water-proof packaging to withstand the harsh conditions of the field. The sensor nodes are manually placed however the mapping to the zones is administered at the WSAN configuration. For energy-efficiency considerations, the sensor nodes are reporting data once per five minutes and were programmed to be in sleep state while not sending or communicating. The data collected by the sensor nodes is gathered by the coordinator node, for local processing and logging. Interaction then is possible between the coordinator node and other devices for managing the delivery of agricultural input according to the decision-making rules.

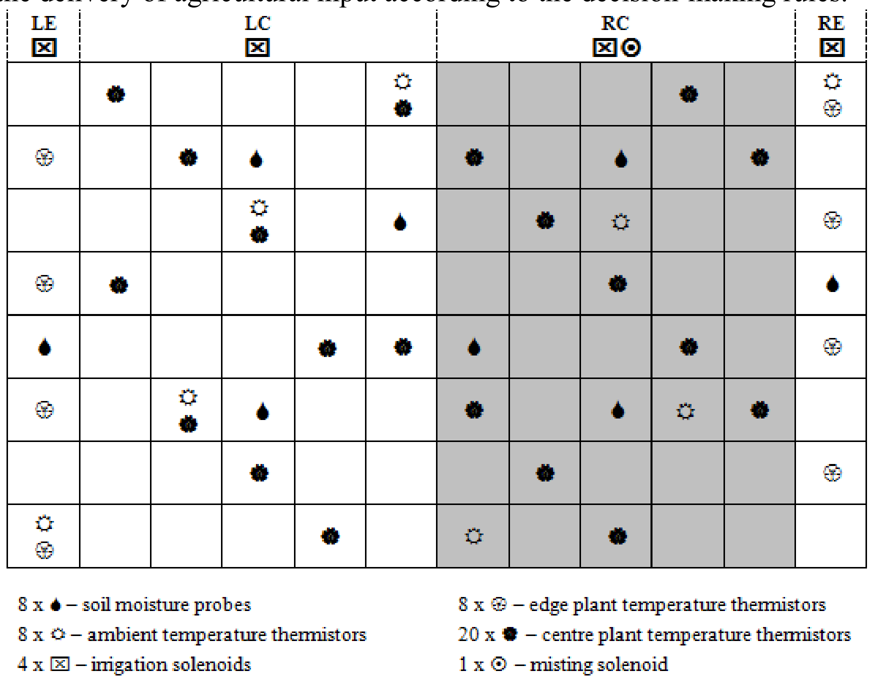

Fig. 15. Schematic indicating the location of the thermistors and soil moisture sensors in the crop array

The application business logic is expressed upon a set of plant/environmental parameters, plant states and possible actions (all defined in the ontology) in the form of a set of rules. Table 2 contains the applicable rules for the RC zone applied during the reproductive phase of the crop. Rules for evaluating the plant states and actions to be performed are shown. According to our model it is possible that different zones or demand sites to have different threshold values depending on different microclimate conditions, different soil types, etc. The threshold values may also vary according to the growing phase of the crop. At least three different phases are defined for strawberry cultivation (i.e., vegetative, reproductive and runner development) and within them several irrigation periods each with different irrigation requirements. In the strawberry fruiting stage, for example, more irrigation is required for quality production. Existing knowledge from the horticulture literature can be easily integrated into our system through the ontology and rule editing tools described earlier.

Two additional parameters must be defined for the prototype to be properly working; the duration of irrigation/misting and an idle time which specifies the amount of time the rules should be disabled, after the action is performed. This is to allow the ecosystem to absorb the changes. The values used for the example application were 1 min and 4 hours respectively.

Table 2. Application rules

Rule	Body
RCDroughtStress	**IF** RCLeafTemperature – RCAmbientTemperature > 0.9 °C
	THEN RCDroughtStress ← **TRUE**
	ELSE RCDroughtStress ← **FALSE**
RCHeatStress	**IF** RCDroughtStress **AND** RCSoilMoisture > 60%
	THEN RCHeatStress ← **TRUE ELSE** RCHeatStress ← **FALSE**
RCNeedIrrigation	**IF** RCDroughtStress **AND NOT** RCHeatStress
	THEN RCNeedIrrigation ← **TRUE**
	ELSE RCNeedIrrigation ← **FALSE**
RCNeedMisting	**IF** RCDroughtStress **AND** RCHeatStress
	THEN RCNeedMisting ← **TRUE**
	ELSE RCNeedMisting ← **FALSE**

6 Discussion

On the agronomic part of the experiment the instrumentation of the strawberry field with the wireless sensor network and the plant-driven irrigation resulted in a notable reduction in water consumption (~20%) with respect to traditional agricultural practices involving user defined timed irrigation based on rules of thumb (twice or thrice a week for 1-2 hours). The later was applied in a parallel setup for the same growing period of the crop. The irrigation treatments were imposed from the beginning of the flowering to the end of the fruit maturity from early June to late September. The quality of the crop was also slightly improved with the precision irrigation treatment vs. traditional irrigation (yield of 686.4 g/plant vs. 679.6 g/plant). Fig. 16 illustrates the water supplied to each group.

The use of the ontology for the organisation of concepts and definition of operational semantics has been successfully tested and revealed the advantages of this approach. Using ontology for defining application business logic emphasizes system flexibility and adaptability. In that sense, our system architecture can be regarded as a reflective architecture that can be adapted dynamically to new requirements. By specifying the rules structure and semantics in an ontology that defines various parameter/states types as well as the arguments that the rules are based upon we can use the ontology to verify rules validity. This also makes easier the inclusion of environmental/plant context parameters in rules, since we know the rules structure and the kinds of values different arguments can take.

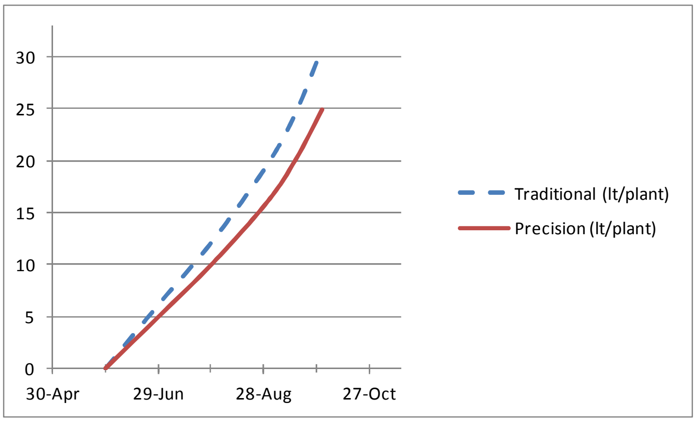

Fig. 16. Total amount of water (lt/plant) supplied to each group

Fig. 17 shows the temperature differential between the thermistors with the 0.9°C threshold marked across the graph. As can be seen from the graph, when the plant overheated and the differential between thermistors crossed the threshold, the pumps were triggered which brought about a reduction in leaf temperature. This graph also demonstrates the reliability of the system control over an extended time period.

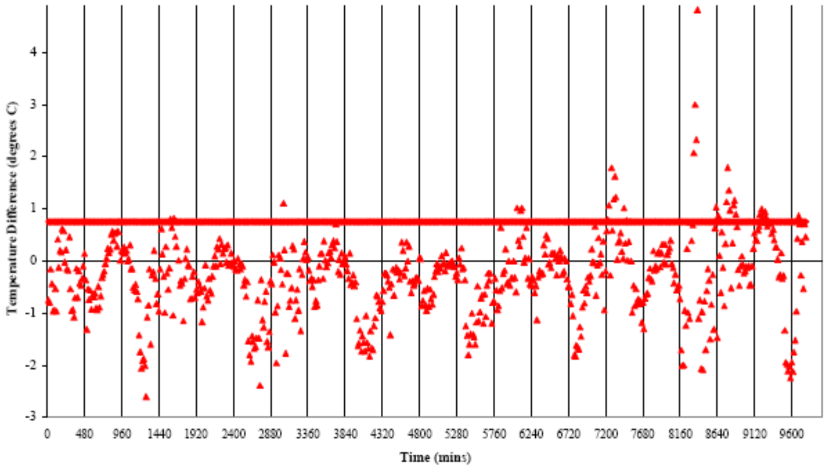

Fig. 17. Zone specific temperature differential between leaf and ambient

The reliability of the WSN is of great importance as lost of data may venture the decision support layer of the system and thus the correct delivery of inputs. There are several measures that have been taken to alleviate this risk. First each sensor node will store each measurement in its local memory and will overwrite it when an acknowledgement is received. In addition the use of sequence numbers in the packets allows the coordinator node to detect easily lost packets, if the MAC-layer fails to deliver them after attempting a number of retransmissions.

7 Conclusion

We have proposed a system that integrates agricultural data measurement and analysis process with the decision making process for determining when and where to irrigate, and how much water to apply. The deployment of smart water management on a large scale is extremely important given the irrigation needs of the agricultural sector (irrigation uses up to 80% of total water in some regions) and the decreasing availability of water for irrigation.

Moving our research towards to a more autonomous system with self-adaptation and self-learning characteristics, we have been exploring ways of incorporating learning capabilities in the system. Machine-learning algorithms can be used for inducing new rules by analysing logged datasets to determine accurately significant thresholds of plant-based parameters and for extracting new knowledge and extending the system ontology.

Acknowledgment

Part of the research described in this paper was conducted in the PLANTS project (IST FET Open IST-2001-38900); the author wishes to thank his fellow researchers in the PLANTS consortium for their input and support.

References

1. Baggio A (2005) Wireless Sensor Networks in Precision Agriculture. In: Proc. ACM Workshop Real-World WSNs, Stockholm, Sweden.
2. Beckwith R, Teibel D, Bowen P (2004) Report from the Field: Results from an Agricultural Wireless Sensor Network, In: Proc. 29th IEEE International Conference on Local Computer Networks, Tampa, FL, USA, pp 471-478
3. Burrell J, Brooke T, Beckwith R (2004) Vineyard computing: sensor networks in agricultural production. IEEE Pervasive Computing, 3(1): 38-45
4. Campbell CS (2004) Calibrating ECH2O soil moisture probes. Application note. Available at: www.decagon.com. Accessed on 21 September 2011
5. Cardell-Oliver R, Kranz M, Smettem K, Mayer K (2005) A Reactive Soil Moisture Sensor Network: Design and Field Evaluation. International Journal of Distributed Sensor Networks, 1:149-162
6. Chatzigiannakis I, Mylonas G, Kokkinos P, Akribopoulos O, Logaras M, Mavrommati I (2011) Implementing multiplayer pervasive installations based on mobile sensing devices: Field experience and user evaluation from a public showcase. J. Syst. Softw. 84(11):1989-2004
7. Crossbow Technology Inc., Mica2 Motes Specifications. Available at: http://www.xbox.com/. Accessed on 21 September 2011
8. von Eicken T, Culler DE, Goldstein SC, Schauser KE (1992) Active messages: a mechanism for integrating communication and computation. In: Proc. 19th Annual International Symposium on Computer Architecture, Queensland, Australia, pp 256–266
9. Estrin D, Culler D, Pister K, Sukhatme G (2002) Connecting the physical world with pervasive networks. IEEE Pervasive Computing, 1(1): 59-69

10. Gay D, Levis P, von Behren R, Welsh M, Brewer E, Culler D (2003) The nesC language: A holistic approach to networked embedded systems. SIGPLAN Not., 38(5): 1-11
11. Goumopoulos C, Kameas A, Oflynn B (2007) Proactive Agriculture: An Integrated Framework for Developing Distributed Hybrid Systems. In: Proc. 4th Ubiquitous Intelligence and Computing, Hong Kong, China, pp 214-224.
12. Guinard A, McGibney A, Pesch D (2009) A wireless sensor network design tool to support building energy management. In: Proc. 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, New York, NY, USA, pp 25-30
13. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: 33rd Annual Hawaii International Conference on System Sciences, Maui, Hawaii, USA, pp 3005-3014
14. López Riquelme JA, Soto F, Suardíaz J, Sánchez P, Iborra A, Vera JA (2009) Wireless sensor networks for precision horticulture in Southern Spain. Comput. Electron. Agric., 68(1): 25-35
15. Mannini P, Anconelli S (1993) Leaf temperature and water stress in strawberry. Acta Horticulturae, 345: 55-61.
16. Pierce FJ, Elliott, TV (2008) Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. Comput. Electron. Agric., 61(1): 32-43
17. TinyOS, Available at: http://www.tinyos.net. Accessed on 21 September 2011
18. Uschold M, Gruninger M (1996) Ontologies: Principles, Methods and Applications. Knowledge Eng. Rev., 11(2):93-155
19. Wang N, Zhang N, Wang M (2006) Review: Wireless sensors in agriculture and food industry-Recent development and future perspective. Comput. Electron. Agric., 50(1): 1-14.