

# USING UBIQUITOUS COMPUTING TECHNOLOGY TO REALISE SCALABLE INTELLIGENT AGRICULTURAL ENVIRONMENTS

Panayiotis Stamatis<sup>1</sup>, Christos Goumopoulos<sup>1</sup>, and Achilles Kameas<sup>1,2</sup>

<sup>1</sup>Research Academic Computer Technology Institute, Research Unit 3, Design of Ambient Information Systems Group, Patras, Hellas  
{stamatis, goumop, [kameas](mailto:kameas@cti.gr)}@cti.gr

<sup>2</sup>Hellenic Open University, 23, Sahtouri Str, Patras, Hellas  
kameas@eap.gr

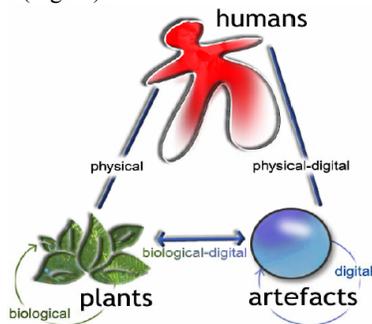
**Keywords:** precision agriculture, Aml, machine learning, intelligent environment.

## Abstract

Wireless networks allow the deployment of sensing systems and actuation mechanisms at a much finer level of granularity than has been possible before. This paper is focused on connecting sensor data with actuators through a decision-making layer with learning capability. The decision making process regarding the provision of agricultural resources is extended this by on-line monitoring significant plant and environmental parameters and by applying machine-learning algorithms for inducing rules by analysing logged datasets to determine the significant thresholds of plant-based parameters.

## 1 Motivation

Currently, there are few discussions on the integration of the natural environment into pervasive computing applications. In this article, we discuss how ambient intelligence technology could be used to encompass plant requirements, to realize intelligent agricultural environments by establishing a three-way interaction between plants, people (end-users and domain experts) and objects (Fig. 1).



**Fig. 1 Hybrid systems with a three-way interaction between plants, people and objects**

Wireless networks allow the deployment of sensing systems and actuation mechanisms at a much finer level of granularity, and in a more automated implementation than has been possible before. Sensors and actuators can be used to precisely control for example the concentration of fertilizer in soil based on information gathered from the soil itself, the ambient temperature, and other environmental factors. Incorporating feedback into the system through the use of sensors, actuators, and adaptation algorithms will allow a more fine-grained analysis that could adjust flow rate and duration in a way that is informed by local conditions. One can imagine the use of such precise information in particularly sensitive high value crops such as wine grapes, citrus fruit and strawberries. This agricultural treatment is known by the term precision agriculture [1].

The interaction of artefacts and plants in the digital space entails the triggering of decision-making procedures. For example, upon determining the local state of a plant a decision may be required for an action to be followed. In the case of an artefact (e.g., a lamp or valve) the local decision-making (or resource management) mechanism resolves conflicts when multiple augmented plants (through superimposition of a technological layer and henceforth mentioned as ePlantations) request a common resource (e.g., light or water). Distributed mechanisms can also be considered to alleviate similar situations, when ePlantations and related artefacts are coordinated for detecting/maintaining a global state/objective in the context of a group of distributed nodes.

This paper is focused on connecting sensor data with actuators through a decision-making layer with learning capability. The decision making process regarding the provision of agricultural resources is originally based on a rule-based knowledge obtained either from the literature or from expert/domain knowledge through a manual and time-consuming characterization process. We extend this by on-line monitoring significant plant and environmental parameters and by applying machine-learning algorithms for inducing rules by analysing logged datasets to

determine the significant thresholds of plant-based parameters.

The remainder of the paper is organised as follows. Section 2 discuss the rationale behind this work and discuss other similar endeavours with an emphasis to the application of data mining techniques. Section 3 describes the basic modules of the system architecture developed for on-line monitoring, logging and proactively responding to the needs of a plantation according to the rules that are stored in its knowledge base. An analogy with a context management process is also identified. Towards to a more autonomous system with self-adaptation and self-learning characteristics, we have explored ways of incorporating Machine Learning aspects in the system as discussed in section 4. Finally the conclusions of our work are presented.

## 2 Problem Statement and related work

The complexity of the parameters/signals to be monitored and controlled in an agricultural environment, coupled with the possible imprecision of the information delivered, makes hybrid system context too complex to be pre-programmed as a fixed set of parameters and rules. A first step towards alleviating this problem is to replace the typical, explicitly coded actions to situations and conditions (which can only prescribe a fixed set of variables) with a multi-level and more knowledge-intensive decision-making framework coupled with reasoning under uncertainty and machine learning techniques. In our approach we apply machine-learning algorithms (discussed in section 4) for inducing rules by analyzing logged datasets to determine the significant thresholds of plant-based parameters.

A research question is to discover the criteria under which individual plants are grouped into the same ePlantation. The most obvious criterion is spatial proximity, as one anticipates that neighboring plants will have similar requirements for resource usage. Nevertheless, the approach allows us to group plants by applying other proximity criteria, such as state of evanescence, age, health status etc, and allowing adaptive resource allocation according to context specific needs.

There are several applications of data mining to particular agricultural problems and issues. An early application considered a program (AQ11) to identify rules for diagnosis of soybean diseases by initially exploiting expert data (questionnaires) and using similarity-based learning [6]. The resulting rules outperformed the expert collaborator rules who eventually adopted the discovered rules in place of his own. In [5] another application dealt with the dairy herd culling, initially exploiting a Livestock database. Domain expertise is considered essential to complement data transformation and machine learning processing skills required to prepare and process the data sets.

Considering the sensitivity of plants on the changing climatic conditions, weather imponderables, pests etc, a system must be flexible and quick responding. In [12], this complexity/uncertainty is overcome by using fuzzy

controllers for the sophisticated control of agricultural systems.

The above mentioned applications are based on the model of exploiting previous mined information offline so as to produce classifications, rule sets or clusters. New (and perhaps more precise) results are provided using the same method and adding additional data to the gathered information. This offline trend inherits the disadvantages of an offline learning scheme. On the other hand, self /online- learning techniques seem to be adaptive and reliable in dynamic environments such as an agricultural farm..

Although not being a direct agricultural application, in [8], the industrial issue of the production line part of classification of agricultural products, a neural network or fuzzy logic based technique outperforms standard template matching and fixed object modeling techniques.

According to [7], sensing precision is related to system monitoring precision. In order to achieve that, the system not only must be equipped with proper sensing components but it must also ensure that this sensing capability is sustained over time, taking under consideration the calibration problem. A prominent example of work related on energy resource sustenance, automatic sensor calibration (on both configuration and dynamic response phases) and finally system tolerance (on both sensor failures and inappropriate development) is the PlantCare system [4].

## 3 System Description

In the past we presented the enabling infrastructure and tools that are used to augment a plant and enable it to interact with AmI devices and artefacts [3]. The work presented in this paper extends the scope of this research and applies ubiquitous computing technology to support resource management in a plantation, which is considered as a collection of individual plants. In this case, the approach of turning each plant into an ePlant and then use the plug/synapse model [9] to achieve collective resource management does not scale well, due to the large number of resulting interactions among ePlants and artefacts. Since considering each plant individually does not yield great benefit, we propose to group plants into ePlantations and enable each ePlantation to communicate as an individual with artefacts. In this way, we can design a scalable system by allocating a set of resources to each ePlantation and then use ePlantation-OS to collectively manage ePlantations. Instead of using a number of devices coupled with each plant, a careful topological distribution of the devices among an area covered with these plants could result a relevant and satisfactory feedback to the system.

In Fig. 2 a graphical representation of an ePlantation is presented: an area containing  $n$  plants is covered by  $m$  Motes, each of which uses a versatile data acquisition board (DAB). Motes are distributed autonomous devices, equipped with a radio transceiver or other wireless communications device (in this case a two-way ISM band radio transceiver), a fully programmable small microcontroller, flash memory for over-the-air-

programming and data logging and finally an energy source (usually a battery). Each DAB can carry a number of different sensor devices, thus, it can be shared among the plants of an ePlantation.

Each Mote communicates with a Mote Gateway. The Mote Gateway allows for the aggregation of the mote network data on an IPAQ (can be any other standard platform like a PC), on which ePlantation-OS is running.

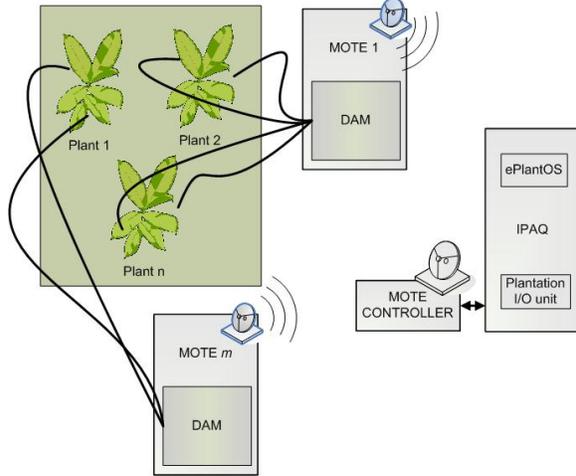


Fig. 2: graphical representation of the ePlantation module

In our approach, an application is realised through the cooperation of ePlantations with artefacts in the form of established logical communication links between services and capabilities offered by the artefacts and the states and behaviours inferred from the plants (in each case services/states are provided through access points called plugs). The plug/synapse model provides a conceptual abstraction that allows the user to describe mixed society ubiquitous applications. To achieve collective desired functionality, one forms synapses by associating compatible plugs, thus composing applications using artefacts and ePlantations as components Figure 3 illustrates, by using a puzzle metaphor, the example application logic in terms of associations made between plugs of an ePlantation and plugs of the eIrrigationSystem, according to the plug/synapse model. When the RCNeedIrrigation plug will be set, for example, solenoid-3 plug will drive the corresponding solenoid to turn on the irrigation for only the RC zone (an ePlantation instance).

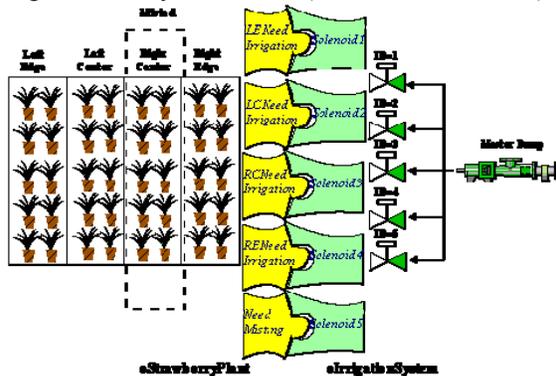


Fig. 3 Schematic illustration of the example application

### 3.1 Context Management Process

At a high level, the process performed by the distributed system presented in Fig. 2 can be viewed as a plant/environmental context management process. We model this process as a measurement-translation-reasoning-actuation control cycle. A mechanism for low-level context acquisition, which reads plant/environmental signals from sensors, starts this cycle. Signals range from selected electromagnetic wavelengths through to volatile organic molecules. This information is probably not initially 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 format. This signal conditioning phase is usually performed within the motes using specialized Analog to Digital conversion circuitry, implemented in such a way as to optimize both network data throughput and system 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 homogeneous or heterogeneous sensors. The determination of photo-oxidative stress, for example, requires monitoring of chlorophyll fluorescence in conjunction with ambient light level signals so as to adjust supplementary light levels. The aggregation of context is an operation that is performed at the higher levels of the system, usually at the IPAQ node.

Having acquired the necessary context we are in a position to assess the state of the plants and decide appropriate response activation. Adopting the definition from Artificial Intelligence, a state is a logical proposition defined over a set of context measurements [11]. This state assessment will be based on a set of rules, which are either obtained from plant science research as part of a time consuming and labor intensive manual process, or as part of a more advanced scheme by utilizing learning capabilities within the system. The low (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 to turn on a light, or to send a message to the user, or a composite one such as a request to add water directly to the soil in the pot in case of water stress, or as spraying mist in case of heat stress. This means that the system has to differentiate between the two kinds of water stress and evaluate the appropriate response. Such a decision may be based on local context or may require context from external sources as well, e.g., a weather station supporting prediction of plant disease spreading.

### 3.2 Basic Software Modules

The deployment of mixed societies composed of plants and artefacts is supported by ePlantation-OS which is a target application modification of the GAS-OS [9]. The ePlantation-OS is additionally composed of the following modules:

**Plantation state diagnosis module:** responsible for the local assessment of the state of each ePlantation. This module applies a normalization algorithm to all local states so as to determine the global state of the ePlantation. Then, a set of actions may be carried out at the ePlantation level (e.g., application of water, increase in light intensity). A rule-based knowledge base is required for the reasoning process;

**Machine Learning(ML) module:** This module runs a ML scheme that uses different ML techniques (clustering, classification) so as to produce more {plant or environment}-oriented rules that will benefit to the adaptation of the system and increase the precision of the system.

**Interaction module:** it implements the interaction scheme between plants and artefacts, in the form of the Plug/synapse model [9];

**The ontology manager** is responsible for the interaction of the nodes in a mixed society of plants and artefacts under the definitions of terms and relationships found in the PLANTATION ontology and for the management of this ontology. Key knowledge encoded in the ontology is the characterization information that relates a plant's status to the measurements provided by one or more sensor systems in the form of given threshold values or range of values or rules. This knowledge is required for the plant behaviour/state assessment

The outline of the ePlantationOS architecture is shown in Figure 4.

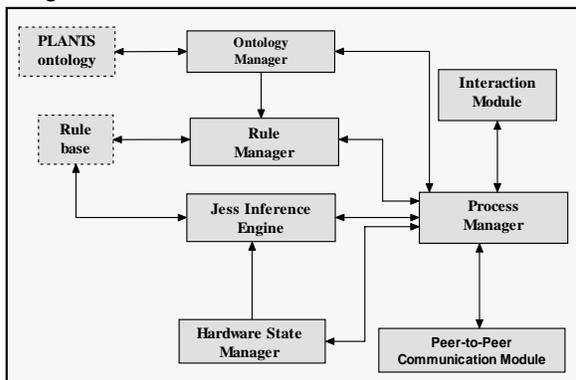


Fig. 4 ePlantation-OS architecture outline

The ePlantationOS encompasses a Communication Module, a Process Manager, a Hardware State Manager, an Ontology Manager a Rule Manager and the Jess Inference Engine. The Communication Module [13] is responsible for communication between different ePlantation/Artefact nodes. This module implements algorithms and protocols for wireless, connectionless communication (using the 802.11b/g protocol) as well as mechanisms for internal diffusion of information

exchanged. The Process Manager is the coordinator module of ePlantationOS. Some of its most important tasks are to manage the processing policies of the ePlantationOS, to accept and serve various tasks set by the other modules of the kernel and to implement the Plug/Synapse model. The Hardware State Manager is a repository of the hardware environment (sensors/actuators) inside ePlantationOS reflecting at each particular moment the state of the hardware. Through the Ontology Manager ePlantations/Artefacts can obtain context-awareness and manifest higher-level behaviour [14]. Applications state their resource or service needs through concepts that are part of the artefact's ontology. The decision-making process is based on a set of rules in operational representation forms that are applied on existent knowledge and allow the use of the ePlantation ontology for reasoning providing inferential and validation mechanisms. The Inference Engine is the module of the ePlantationOS that supports the decision-making process. This module exploits the Jess rule engine (*Java Expert System Shell*) [15]. The execution of this module is started based on the initial facts (defined by the Rule Manager from knowledge emerged from the PLANTS Ontology through the Ontology Manager) and the rules stored in the rule base in Clips format.

The Inference Engine module is informed for all the changes of parameters' values from sensor's measurements through the Hardware State Manager. When the Inference Engine is informed for such a change it runs all its rules. When a rule is activated the Inference Engine informs for the activation of this rule and for the knowledge that is inferred the Process Manager, that is responsible to transfer this knowledge to anyone that needs it. A graphical depiction of the Inference Engine is in Fig. 5.

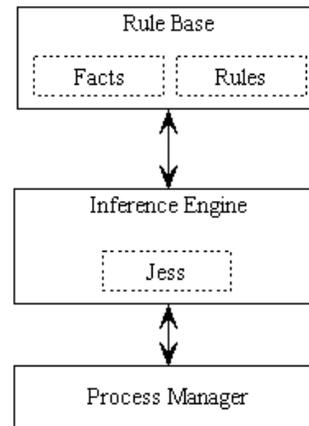


Fig. 5 Inference Engine

### 3.3 System operation

In this subsection we describe the typical operation flow of the ePlantation system. Starting, each plant of a set of  $m$  plants is monitored by a set of sensing devices. These input devices allow the system to receive precision data, which is crucial for correct plant signal translation.

Precision agriculture systems require the remote detection of exogenous plant signals (intrinsic signalling compounds are not considered because the insertion of sensors within the plant tissue not only wounds the plant but provides an effective entry point for pest and disease). A range of plant signals has been considered to indicate the plant input requirements. The evaluation of plant signals for development was carried out using the following criteria:

- Is the signal produced in accordance with the plant stress/ need?
- Do COTS sensors exist for the signal?
- Do COTS sensors exist that detect the signal in the range emitted by the plant?
- How rapidly is the signal produced following the initiation of the stress/need?
- Can the plant signal be analysed to determine threshold levels for actuator induction?
- Can COTS actuator systems remediate the plant requirement?

The test case presented in this paper considers the following set of sensing devices: *fluorescence meter*, *ambient light*, *thermistors* and *soil moisture* sensor. These devices output five low level values: chlorophyll fluorescence, light PAR, ambient (environment) temperature, plant temperature and soil moisture value. The chlorophyll fluorescence (Electron Transport Rate parameter) acts as a general plant health indicator and provides a reference for the machine learning process to be applied. The mote Gateway proceeds in real time the data received from every mote towards the main processing unit the ePlantation-OS is running on.

Using the chlorophyll fluorescence value, the ETR high level attribute is computed. This attribute gives a clear estimation about each plant stress condition. According to this estimation, the system can compute a Boolean “isHealthy” attribute representing the state of each plant. The state of the ePlantation is then estimated using all data recently received (Plantation state diagnosis module). This estimation is done by applying a mean value calculation to every mote’s acquired data set. In general, this is analogous to estimating an average value of  $n$  ePlants. After estimating the ePlantation overall state, a set of actuation rules is applied. The starting rules of the plant assessment and treatment as used in this work is stated below:

```
If soil moisture < 60%
Then <request Irrigation>.
```

```
If not HEALTHY
Then <report to plant expert>.
```

The ML module aims to improve the effectiveness of the system a) by reducing resource consumption (water, fertilizer etc) and b) by eliminating the time plants are stressed in any way. This target can be achieved by creating new plant assessment rules using other context information (e.g. plant/ambient temperature) and then adapting the rule base (especially the actuator action set) to these changes.

## 4 Machine Learning

In order to apply machine learning, a java invoker of the WEKA core was set on the ePlantationOS. In general, the New Zealand-developed WEKA (Waikato Environment for Knowledge Analysis) [10] is a collection of machine learning algorithms written in Java. Weka is an open source software providing user extendable algorithms, making the tool well-suited for developing new machine learning schemes. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

The first step in using the workbench is to invoke the workbench conversion programs to prepare data (usually provided in spreadsheet or database format) into a file format which all of the different machine learning techniques recognise. It is then possible to run a variety of machine learning schemes on the data and to view and compare the results. The schemes work in many different ways and no single scheme will outperform the others on all datasets. Next, through the Weka core, the dataset is used to construct a classification model which is used by the decision making module.

In general, using the collected data, data mining can be applied either by running a decision making (classification) algorithm either by running a clustering algorithm. The former results a set of rules and the latter results clusters of relation. The rule set can enhance or even replace the obsolete rule set used to decide either a particular state of the ePlantation such as the isHealthy attribute, or the target action rules (such as Irrigation and Misting). On the other hand, clustering based on particular proximity criteria-attributes may lead not only to rule activation threshold refinement but also to detection of new –non obvious- proximity rules. In this paper, we present the results and conclusions after applying the EM clustering algorithm, discussed in the next paragraph.

### 4.1 Dataset

The dataset used for the machine learning application was collected in the context of the PLANTS project [2] where the plant monitored was a strawberry plant at an early development phase. A segment of the dataset, used for running the machine learning algorithms with the WEKA workbench, is given in Table 1.

**Table 1. A segment of the dataset, used for running machine learning algorithms with the WEKA**

ETR	isHealthy	AmbC	Plant C	Moisture
387,15	OK	22,37	21,65	0,65
432,9	OK	22,37	21,65	0,64
412,83	OK	21,65	21,16	0,64
372,36	OK	21,65	20,92	0,64
382,41	OK	21,28	20,80	0,62
363,13	OK	21,28	20,80	0,62
432,95	OK	22,01	21,16	0,62
422,3	OK	22,01	21,16	0,61
435,31	OK	22,01	21,04	0,61

448,31	OK	22,01	20,92	0,61
463,81	OK	22,01	20,92	0,60
422,99	OK	22,01	20,67	0,60
315,88	OK	21,65	20,43	0,59
305,78	Not OK	21,28	20,43	0,59
324,47	OK	21,65	20,92	0,92
436,07	OK	22,74	21,52	0,93
423,33	OK	23,10	21,40	0,93
498,06	OK	22,74	21,65	0,93
534,43	OK	23,46	22,01	0,94
532,53	OK	22,37	21,16	0,94
482,72	OK	22,74	21,40	0,94
508,19	OK	22,37	20,67	0,94
510,98	OK	22,01	20,67	0,94

The column labelled as ETR is calculated by combining the chlorophyll fluorescence and light PAR measurements. The rule then that defines the status of the plant (Healthy, Not Healthy) is as follows:

```
If ETR <= 310
Then <deactivate isHEALTHY>.
```

The above table is translated into the ARFF format, an attribute/value table representation that includes header information on the attributes' data types. It is then possible to run a variety of machine learning schemes on the data and to view and compare the results.

#### 4.2 EM Clustering

The aim of this work is to demonstrate the applicability of clustering to the particular application and not to find the best suitable algorithm for clustering plant properties. Therefore, we used the Expectation Maximization (EM) algorithm, which is a popular statistically formalized method. This algorithm presents good convergence properties and is commonly used for several partitioned clustering applications.

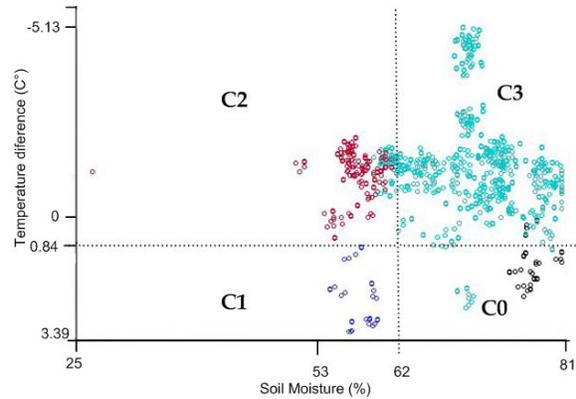
The EM Clustering algorithm was set according to the WEKA predefined parameters (maxIterations=100, minStdDev= 1.0E-6, numClusters=-1, seed=100), according to the (low level) attributes of plant temperature, environment temperature and soil moisture. The high level values of ETR and isHealthy were ignored during the clustering procedure. The result is a four cluster set, with each cluster characteristics presented in Table 2.

**Table 2: Results applying EM clustering using default settings**

Attribute	Normal Distribution. Mean	StdDev
<b>Cluster: 0</b> Prior probability: 0.2486		
AmbientC	20.285	0.9228
PlantC	18.8293	1.9115
SM	0.6044	0.0769
<b>Cluster: 1</b> Prior probability: 0.2837		
AmbientC	20.9822	0.6592
PlantC	19.8459	0.7244

SM	0.7475	0.0424
<b>Cluster: 2</b> Prior probability: 0.0559		
AmbientC	26.2583	1.3866
PlantC	28.4052	1.3866
SM	0.6534	0.1039
<b>Cluster: 3</b> Prior probability: 0.4118		
AmbientC	23.484	0.7129
PlantC	21.9673	1.7046
SM	0.6798	0.0598

Fig. 6 depicts a plot of the data set instances in respect of soil moisture value (axis x) and temperature difference (AmbC-PlantC) (axis y). Axes and values are added for clarity. The dotted line boundaries are placed so as to approximately separate the four clusters. The different colors indicate the cluster a data instance belongs to. The colors of black, blue red and cyan refer to cluster C0, C1, C2 and C3 respectively.



**Fig 6. Clusters in respect of soil moisture and temperature difference**

As the plot depicts, each cluster is generally allocated in one of the four areas separated by the dotted vertical and horizontal lines. What is notable is that cluster 4 contains approximately the 95% of the instances marked as healthy (attribute isHealthy=TRUE). The next step of the procedure was to use expert knowledge to analyze all instances marked as not healthy (attribute isHealthy=FALSE) according to the necessary treatment so as to become healthy.

What was found is that all instances belonging to clusters 1 and 2 were needed irrigation treatment. For cluster 0, 82% of the instances required misting treatment and the rest were all healthy. Finally, 9% of the instances belonging to cluster 3 were requesting irrigation and 3% requesting misting. All other instances were stated as healthy. These facts lead us consider two new high level attributes: HeatStress and WaterStress. According to the cluster approximant boundary lines, we assume the following assessment rules:

```
If soil moisture < 62%
Then <WaterStress is Activated>.
If (AvgPlantC - AmbC) > 0.84°C And Soil
Moisture > 62%
Then < HeatStress is Activated>
```

According to these rules, two plant treatment actions may occur:

```
If <WaterStress is Activated>  
Then <request Irrigation>  
If <HeatStress is Activated>  
Then <request Misting>.
```

What is considerable is that the irrigation treatment rule matches (with a small deviation of 2% approximately) to the starting rule of the plant assessment provided by experts and presented in paragraph 3. Although the abovementioned rules were reasoned by experts as rational to use, further work is needed towards the verification of the above estimation. This includes a) the execution of long haul experimentations to verify repeatedly the assessments and to mask out possible errors or aberrations and b) generalization of the process for different plant variations and different environmental-climatic conditions; a process that could finally lead to the substantiation of the above rules.

## 5 Conclusions and future work

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. In this paper machine-learning algorithms have been used for inducing new rules by analysing logged datasets to determine accurately significant thresholds of plant-based parameters. In a more advanced scheme when a new plant variety is introduced, an automatic characterisation phase could be initiated during which the proper growing conditions will be determined. The chlorophyll fluorescence parameter forms the backbone of the feedback mechanism to determine the photosynthetic efficiency of the plant, and from this determine how productive the plant is under the given conditions. Such an approach is currently under investigation.

To counter with the uncertainty of data work is in progress to define a model describing the uncertain context. Quality indicators can be specified so that the end-user (either an application or a person) can make judgements on the confidence level that the information entails. Uncertain context mechanisms such as probabilistic logic, fuzzy logic and Bayesian networks can be evaluated and applied accordingly.

## Acknowledgements

This paper describes research carried within the PLANTS project (IST FET Open IST-2001-38900); The authors wish to thank their fellow researchers in the PLANTS consortium.

## References

[1] Bouma J., Stoorvogel J., van Alphen B.J., Boontink H.W.G.: Pedology, Precision Agriculture, and the Changing Paradigm of Agricultural Research. Soil

- Science Society of America Journal, 63(6), 1763-1768 (1999).
- [2] PLANTS Consortium: D.5.2 Final version of Style, Middleware and Distributed System, PLANTS IST-2001-38900 Technical Report, March 2006.
- [3] J. Calemis, C. Goumopoulos, A. Kameas "Talking Plant: Integrating plants behaviour with ambient intelligence" , Intelligent Environments' 06, Athens Greece
- [4] LaMarca, A., W. Brunette, D. Koizumi, M. Lease, S. B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. Making sensor networks practical with robots. In Pervasive Computing. First International Conference, Pervasive 2002. Proceedings (Lecture Notes in Computer Science Vol.2414). 2002. Zurich, Switzerland: Springer-Verlag. p.152.
- [5] R.J. McQueen, S. R. Garner, C.G. Nevill-Manning, and I.H. Witten. *Applying machine learning to agricultural data*. Computers and Electronics in Agriculture, 12(4):275--293, June 1995
- [6] Witten I.H., Cunningham S.J., Holmes G., McQueen R.J., Smith L.A. (1993). "Practical Machine Learning and its Potential Application to Problems in Agriculture". In Proceedings of The New Zealand Computer Conference vol. 1 pp. 308-325, Auckland, New Zealand
- [7] Ehret DL, Lau A, Bittman S, Lin W, Shelford T. 2001. Automated monitoring of greenhouse crops. *Agronomie* **21**, 403–414
- [8] Kavdir, I. and D. E. Guyer, 2003. *Apple grading using fuzzy logic*. *Turkish J Agric For*, 27(2003) 375-382.
- [9] Drossos, N., Goumopoulos, C. and Kameas, A., A Conceptual Model and the Supporting Middleware for Composing Ubiquitous Computing Applications, Proc. of the IEE International Workshop on Intelligent Environments, Colchester, UK, 28-29 June 2005
- [10] Waikato Environment for Knowledge Analysis (WEKA), <http://www.cs.waikato.ac.nz/ml/weka/>
- [11] Russell S. and Norvig P, Artificial Intelligence: A Modern Approach. Prentice Hall, 2nd edition, 2003.
- [12] Morimoto T., Hashimoto Y., Hoshi T., An intelligent control technique based on fuzzy controls, neural networks and genetic algorithms for greenhouse automation, in: Kozai T., Murase H. (Eds.), Proc. 3rd IFAC-CIGR Workshop, Artificial intelligence in agriculture, Chiba, Japan, April 1998, pp. 61-66, Pergamon/Elsevier Science, Oxford, UK
- [13] Kameas A., et al, "eComp: An Architecture that Supports p2p Networking Among Ubiquitous Computing Devices", in Proceedings of the 2nd International Conference on Peer-to-Peer Computing (P2P'02), Linköping, Sweden, Sept 2002
- [14] Goumopoulos, C., Christopoulou, E., Drossos, N., and Kameas, A., The PLANTS System: Enabling Mixed Societies of Communicating Plants and Artefacts, A., Proc. of the 2nd European Symposium on Ambient Intelligence (EUSAI 2004), LNCS 3295, pp. 184-195, ISBN 3-540-23721-6, Springer-Verlag, Eindhoven, the Netherlands, November 8-10, 2004.

[15] Jess - the Rule Engine for the Java Platform,  
<http://herzberg.ca.sandia.gov/jess/>