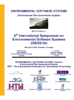
In **Sixth Int'l Symposium on Environmental Software Systems (ISESS-05)**, (Jakeman, A. and Swayne, D. A., Eds.), Sesimbra, Portugal, International Federation for Information Processing (IFIP), May 2005.



Abacus: A multi-agent system for meteorological radar data management and decision support

Ioannis N. Athanasiadis, Marios Milis and Pericles A. Mitkas

Electrical and Computer Engineering Dept, Aristotle University of Thessaloniki, Thessaloniki, Greece

Silas C. Michaelides

Meteorological Service, Nicosia, Cyprus

Abstract: The Meteorological Service of Cyprus operates a Doppler radar at the mountainous region of the island. Data-streams recorded by the radar are used for weather forecasting and, especially, for identifying oncoming precipitation incidents and issuing (potential) warnings. However, the continuous processing and evaluation of radar data requires significant efforts by the meteorologists, both for data processing, storage, and maintenance, as well as for data interpretation and visualization. To assist meteorologists and to automate a large part of these tasks, we have developed Abacus, a multi-agent system for managing radar data and providing decision support. Abacus' agents undertake data-management and visualization tasks, while they are also responsible for extracting statistical indicators and assessing current weather conditions. In addition, Abacus' agents can identify potentially hazardous incidents, disseminate preprocessed information over the web, and enable warning services are provided via email. In this paper, Abacus' agent architecture is detailed and agent communication for information diffusion is discussed. Focus is also given on the fully customizable logical rule-bases used for agent reasoning required in decision-support. The platform has been tested with real-world data from the Meteorological Service of Cyprus.

[*] Send correspondence to: Ioannis N. Athanasiadis (`ionathan@ee.auth.gr`)

1. Introduction

1.1 The meteorological radar 'Kykkos'

The Doppler radar of the Meteorological Service of Cyprus is installed on the northwestern mountainous region of the island. The radar is established near Kykkos medieval monastery; hence, it is named after it. The radar "Kykkos" characteristics are comprehensively shown in Table 1. Kykkos' antenna is able to execute a complete rotation of 360° on the horizontal plane, while changes its vertical target for distinct elevation levels. Kykkos' beam is reflected by the clouds or other obstacles within its range. In this manner, Kykkos radar scans provide a three-dimensional overview of the atmosphere around the island. Kykkos is operated remotely from Larnaka Airport, where the Weather Forecasting Office of the Meteorological Service of Cyprus is locate. The radar may operate in two modes: In the surveillance mode, radar scans are projected on a terminal monitor in real time, while in the off-line mode, radar data volumes are acquired, according to pre-defined scan strategies and consequently are stored in a local hard disk.

Table 1. *Kykkos radar parameters*

Transmitter – receiver	
Peak Power	158kW
Frequency	5.7 GHz
PRF	250 Hz and 1180 Hz
Pulse duration	2μs and 0.7 μs
Antenna	
Diameter	2.5 m
Beam width at half power	1.1°
Power gain	44 dB
Polarization	Horizontal
Data features	
Maximum range used	120 km
Radial resolution	500m
Number of power levels	80 (-15dBz to 65dBz)

1.2. Kykkos radar use and related problems

Typically, Kykkos radar operates in order to assist the Weather Forecasters of the Meteorological Service of Cyprus in very short term forecasting practices. More specifically, radar data are used for:

a. The surveillance of the weather conditions in real time,
b. The identification of precipitation patterns within the area covered by the radar, and
c. The forecasting of extreme events and the issue of related warnings.

Typically, these tasks require the engagement of scientists, who are responsible for acquiring radar's data, preprocessing them appropriately, and ultimately making decisions. This process involves data filtering and restoration, as radar's reflections are disturbed by natural obstacles that cause *beam's occultation* and *ground clutter* problems. These two phenomena (described in

Box 1) have been taken into consideration through the designing of system Abacus, so as to reduce their negative side-effects in the decision-making process. The current settlement of Kykkos radar requires an operator (a human expert) that continuously monitors the radar's reflections and decides upon interesting events/incidents. The continuous processing and evaluation of radar data requires significant efforts by the meteorologists, both for data processing, storage, and maintenance, and for data interpretation and visualization.

Box 1: *Beam Occultation and Ground Clutter disturbance to the radar's reflections.*

Beam Occultation: The radar beam can be blocked by the presence of obstacles (mountains or hills). This causes the alteration of the reflection value of the beam. The beam can be blocked totally or partially. In the later case the radar beam can pass over the mountain, but the measurement is disturbed. Ground Clutter: When the radar beam hits the terrain, an echo occurs. This echo can be considered as rain signal by mistake. This kind of disturb is called ground clutter. The only characteristic that makes the Ground Clutter reflections differ from real reflections is their time permanency (zero velocity). Thus in order to minimize the effect of this phenomenon zero velocity echoes must be neglected. Ground clutter is the reason that the radar detects strong echoes both when the sky is clear and when heavy rainfall occurs.

For supporting meteorologists with the abovementioned tasks, we developed Abacus, a multi-agent system for managing radar data and decision support. Abacus constitutes a middleware software system that intervenes between Kykkos radar and the meteorologists and provides advanced services to the Meteorological Service of Cyprus. Abacus' core functionalities concern:

a. The review, transformation and preprocessing of radar's scans,
b. The identification of weather conditions at real-time and their evolution through time,
c. The provision of information services to authorized personnel and the public using email notifications, distribution of alarms over the internet, etc.

1.3 Problem description

Abacus platform has been designed and developed according to the needs and the requirements of the Meteorological Service of Cyprus. The main objective of this effort was the implementation of a system, which would assist the meteorologists to carry out their duties more easily and effectively. In particular, the Abacus system was envisioned as a platform of autonomous, artificial '*meteorologists agents*' that undertake Kykkos radar data management and exploit appropriately all information that it produces. In order to achieve this goal, Abacus is required to implement the following functionalities:

a. The pre-processing of successive radar scans. This activity involves:
    i. the caption of reflection values produced by the radar, their transformation and filtering (taking under account *ground clutter* and *beam occultation* biases),
    ii. the calculation of various quantitative metrics and qualitative indices within the radar's range, generating valuable meta-data,
    iii. the depiction of data, metrics and indices in a variety of graphic representations, and
    iv. the presentation of raw radar data, generated metadata, and diagrams to the final users, through a friendly and functional user interface.

Note that extracting meta-data (metrics and indices) from raw radar scans constitute (independently from their further use by the system) a demanding, thus valuable, service for the study and the analysis of the related meteorological phenomena.

b. The provision of alerting services whenever predefined conditions are satisfied in specific areas of interest. Users are enabled to define rules for describing various weather conditions, as complex combinations of meta-data metrics and indexes. These rules incorporate both time and space restrictions and each one describes a certain type of incident and is associated with a certain warning. Warnings produced by these rules are subsequently disseminated in three different ways:
    i. By sending an email message to a predetermined list of receivers, containing the conditions, place, and time that raised the alarm.
    ii. By raising a sound-alarm that notifies scientists staying close.
    iii. By posting a warning message on webpage, containing the conditions, place, and time that raised the alarm.

1.4 Methodology adopted

For Abacus development, we used software agents as the building block of both system design and software coding. With the notion of an **agent,** we define a software entity characterized by autonomy, reactivity, pro-activity, and social ability. Certain types of software agents have abilities to infer rationally and support the decision making process. Agent-based systems may rely on a single agent, but the advantages of this initiative are revealed in the case of Multi-Agent Systems, which consist of a community of co-operating agents. Several agents, structured in groups, can share perceptions and operate synergistically to achieve overall goals (Jennings, Sycara and Wooldridge 1999). Developing intelligent software applications for environmental management and assessment with agents is advantageous (Athanasiadis and Mitkas 2004c), as the notion of an agent is easily comprehensible by natural scientists, environmentalists, economists, social scientists and software engineers. In this way, value conflict problems can be tackled efficiently. In addition, agent technology benefits for rapid prototyping and software reusability can be also considered as further advantages of such an approach. Motivated by these findings and our prior experience in environmental informatics, we followed an agent-based approach for the design and implementation of the Abacus platform. In the following section, we present the abstract architecture of the system and all agent types that constitute the platform.

2. The Abacus platform

2.1 Abstract architecture and services provided

The Abacus platform as a middleware for providing information services is illustrated in Figure 1. Reflection raw data produced by Kykkos radar are captured by the platform and are transformed for supporting data management, web notification, email warning and sound alarm services to the final users. Abacus users are clustered in three distinct groups:

(a) the radar operator, who administrates the system,
(b) the Meteorology Service personnel, who has full access to the raw data, generated meta-data and all graphical representations, and notification/alarm services,
(c) the indirect users (the people), who are granted access to the generated information via web-based information services.

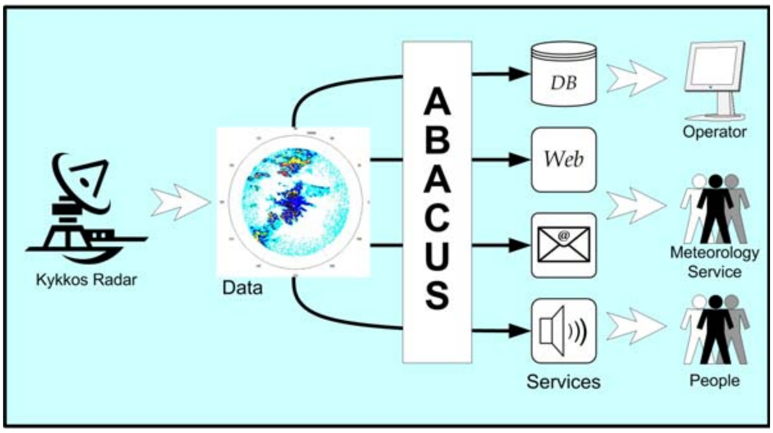


Figure 1. Abacus abstract architecture: a middleware for providing information services

2.2 Platform design

The Abacus functionality described in the previous section has been realized through a layered agent-based architecture. Athanasiadis and Mitkas (2004b, 2005a,b) have introduced a three-layer agent architecture for developing environmental applications with software agents. A similar approach was followed for Abacus design. Specifically, system activities have been clustered in three functional, cooperative layers:

i. The ***contribution layer***, that is responsible for acquiring radar scans and data filtering pre-processing activities.
ii. The ***management and processing layer***, where meta-data (metrics and indices) are generated and decision rules are applied for identifying potentially interesting incidents.
iii. The ***distribution layer***, that implements interfaces with the system operator and disseminates warnings, via email, web, and sound signals.

In each layer, generic agent roles have been appointed for realizing system functionalities. Agent roles correspond to specific agent behaviours, which have been realized in six generic agent types. Abacus generic agent types are:

(a) The **radar agent**, which retrieves radar scans and restores data biased by beam occultation and ground clutter effects.
(b) The **meteorologist agents**, which form a community of cooperating agents, each one of which is responsible for an annular sector within the radar's range. Each one calculates metrics and indices (i.e. extracting meta-data) within its sector and applies decision rules for assessing the weather conditions and issuing alarms at a local level.
(c) The **Abacus agent** summarizes all information extracted by the meteorologist agents. It is also responsible for issuing warnings at a global scale.
(d) The **GUI agent** implements the graphical user interface with the direct user. It is responsible for instantiating the platform at start up and for preparing and visualising graphs and maps on the end user terminal.
(e) The **DB Agent**, that connects with the meteorological database, and stores radar data and extracted meta-data.

(f) The **Alarm agent**, which is responsible for disseminating the alarms generated (either in local or global level) via email, web or sound warnings.

The Abacus platform architecture and the synergies between the three layers are illustrated in Figure 2.

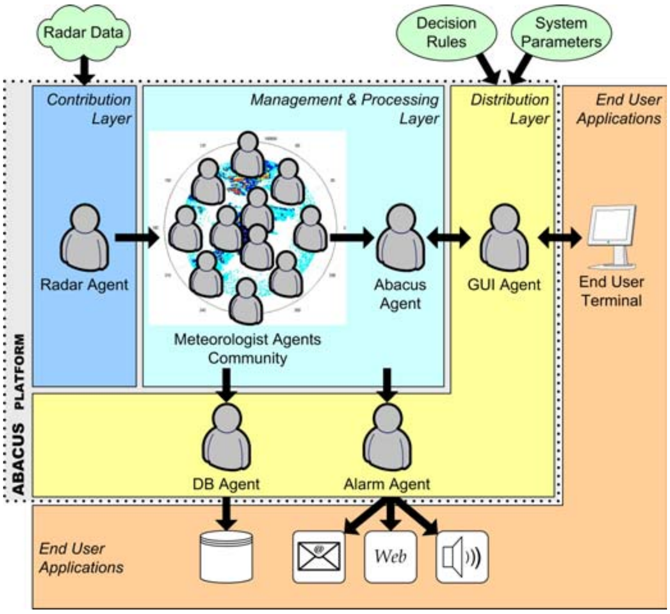


Figure 2. Abacus Platform Architecture

2.3 Meteorologist agent community and sectors of responsibility

As indicated above, each radar scan is made up of a reflection table sized 240×360. The 240 table lines correspond to the 240 actinic steps (each one of which is 500m) and the 360 columns to the 360 sectors (each one of which is 1deg) recorded in every radar rotation. The total surface covered by the radar is segregated to the community of meteorologist agents, each one of which is responsible for an annulus sector. Each agent is identified by its polar coordinates. In Figure 3, an example of a hundred agents (distributed 10×10) on the plane is illustrated: 10 agents per annulus and per sector. Obviously, each agent's area of responsibility corresponds to an annular sector and is composed of a matrix of 24×36 values. In Figure 3a, agents' sectors of responsibility are depicted. The outer annulus is covered by ten agents, and the eleventh agent will be situated precisely under the first one. Note that all agents access the same amount of information (a matrix of 24×36 values in this example), but the surface of the area of responsibility differs. Figure 3b shows the information matrix for the first agent.

2.3 The meteorologist agent community and decision-making

Agent reasoning and on-line decision-making is a key feature of the platform. Each one of the Meteorologist agents incorporates its own rules for assessing weather conditions within its area of responsibility. All decision rules are defined through the platform GUI by the meteorologists at the Meteorological Service of Cyprus. Each rule consists out of three parts:

(a) An *assumption antecedent* that relates meta-data parameters (indices) with certain value ranges.

(b) A *time constraint* that defines a time interval within the assumption antecedent should be satisfied.

(c) The *corresponding alarm*, which is the consequence (decision) in which will the agent will conclude whenever the assumption is satisfied for the corresponding time constraint.

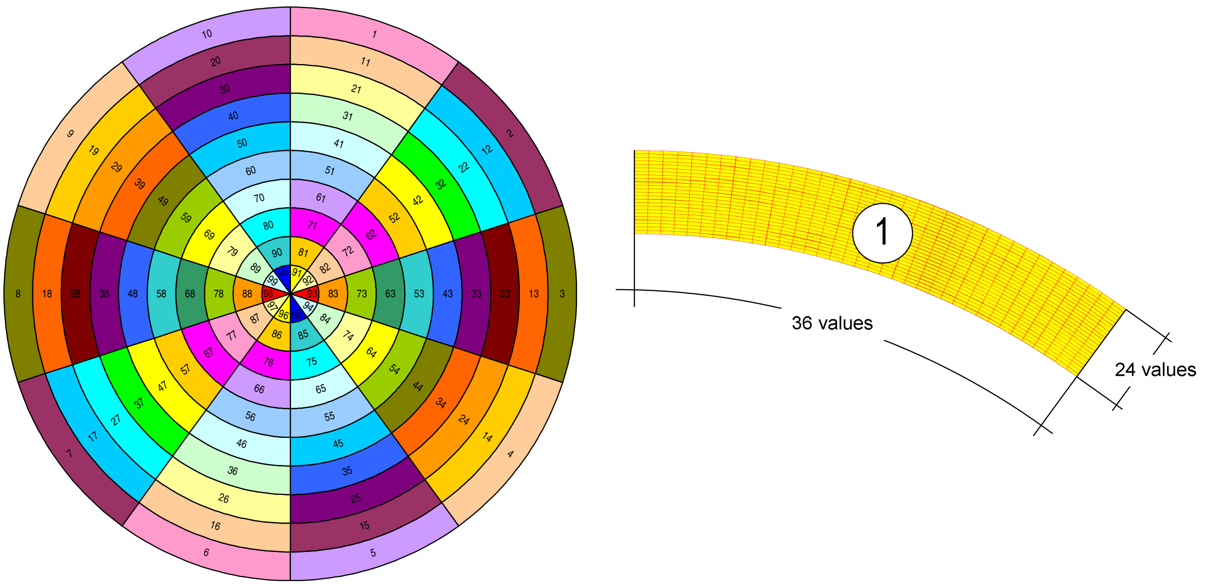
For shaping complex rules, the user is enabled to prescribe reflection value ranges that correspond to *cloud types* and use them for defining various assumptions. Four indices are calculated by Meteorologist agents and the Abacus agent, for their area of responsibly. These indices are used for building a rule's assumption:

i. Mean reflection value within the annular sector.
ii. Mean reflection value per cloud type within the annular sector.
iii. Surface coverage percentage per cloud type within the annular sector.
iv. Percentage of cells per cloud type within the annular sector.

A simple rule that could be constructed and embedded into a meteorologist agent is the following:

```
if percentage of cells within range 50-60dBz is greater than 60%
for 3 succeeding scans
then raise a sound alarm
```

The parameterization of the decision rules and the selection of these indexes were based on an in-depth study on radar data use and radar characteristics and analysis (Michaelides et al. 2003). Note that the user can associate several rules to each agent. Each rule is fired separately and independently from the rest ones. In this way, a fully customizable rule set is embedded in each agent for supporting the decision-making process of the system.



(a) (b)

Figure 3. (a) Sectors of Responsibility of each Agent, and (b) Information matrix for Agent 1

3. ABACUS Implementation

3.1 Software design and Ontology

ABACUS model was designed using the GAIA methodology (Wooldridge et al., 2000). The software agent interaction has been specified using the Agent–Object–Relationship Modeling Language (AORML), introduced by Wagner (2003). Figure 4 illustrates the AORML external agent diagram, which defines all communication among Abacus agents and users.

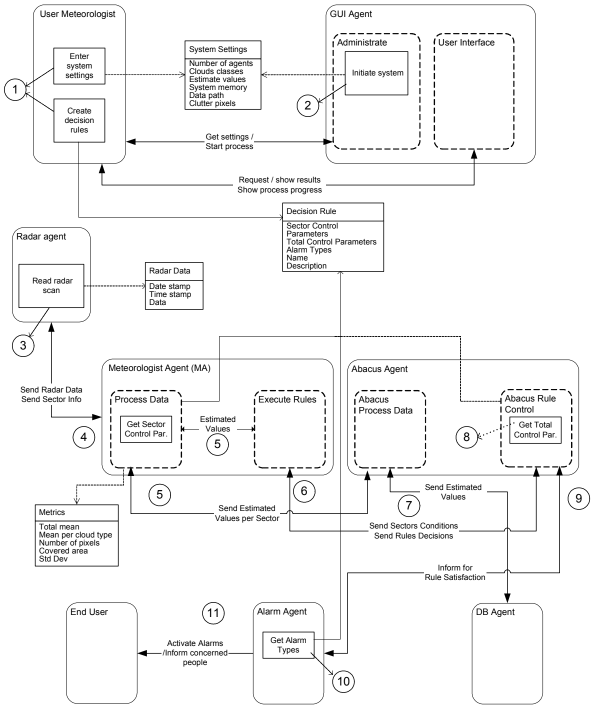


Figure 4. AORML External Agent Diagram

3.2 System operation

Abacus operates in eleven steps, shown Figure 4 and detailed below:

Step 1:	The Application User defines the system settings and selects the decision rules.
Step 2:	The GUI Agent initiates all platform agents at runtime, based on user settings.
Step 3:	The Radar agent reads radar data and filters / preprocesses them.
Step 4:	Filtered radar scan is sent to the Meteorologist Agents
Step 5:	Meteorologist Agents calculate indexes and metrics for their area of responsibility and fire their decision-making rules.
Step 6:	All meteorologist agents send the extracted meta-data and alarm to the Abacus agent
Step 7:	Abacus agent concatenates data from all annular sectors and creates a joint meta-data view on the current scan. This view is forwarded to the DB Agent in order to be stored.
Step 8:	Based on the joint data view and local alarms, the Abacus agent activates its decision rules and fires alarms on a global level.
Step 9:	The Abacus agent forwards all raised alarms to the Alarm agent.
Step 10:	Alarm agent processes the raised alarms, (creates webpage or email content).
Step 11:	Alarm agent activates the alarms, via web, email, or sound signal.

3.3 Agent communication and Abacus ontology
Information communicated by platform agents is in a structured form. Agent messages follow a generic ontology developed by using the Protégé–2000 ontology editor [Noy et al., 2001]. Part of the Abacus Ontology developed is shown in Figure 5, where the concepts of the system, along with agent Actions and Predicates are depicted. The slots of the various concepts have been configured in order to contain the appropriate content communicated by the agents.

3.4 Implementation Details
The Abacus platform was implemented in Java. JADE suite has been used for agent development [Bellifemine et al., 2001]. Agent design and implementation in Abacus conforms to the FIPA specifications [FIPA, 2002]. Note that the Abacus user is required neither to have any programming skills, nor to understand the internal functionalities of the platform. The advantages of the implemented system are its user-friendly interface ant its open, easy-to-parameterize implementation.

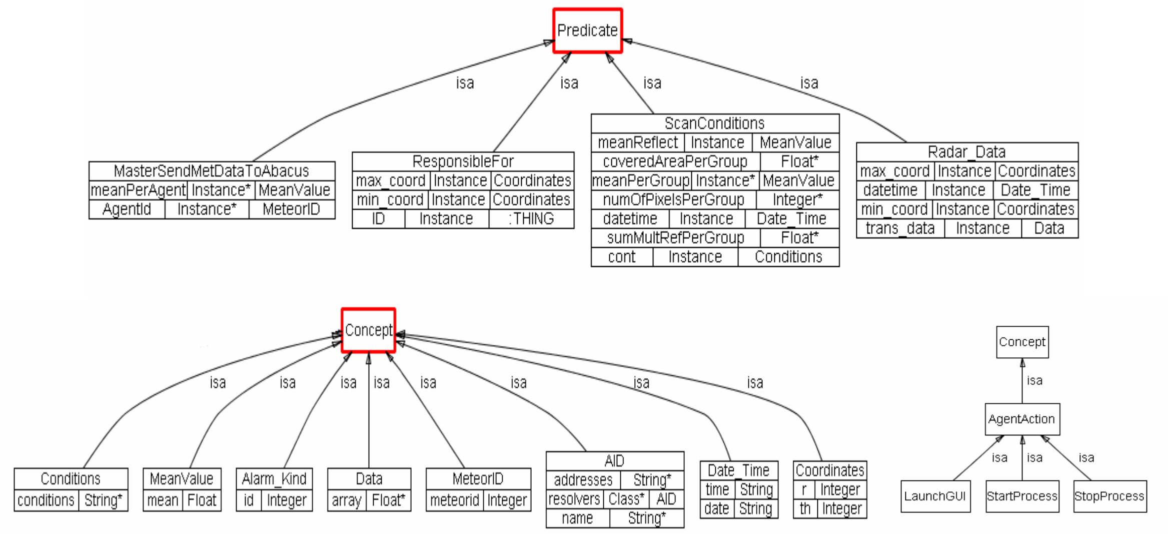


Figure 5. Abacus Ontology: Concepts, Predicates and Agent Actions

4. Demonstration and future work
The Abacus platform has been demonstrated to the Meteorological Service of Cyprus. Currently, Abacus operates on an off-line mode (while the surveillance mode is also enabled) and handles real data scans at operational time frame (1-2 minutes per scan). A screenshot of the main GUI is shown in Figure 6.
Abacus system constitutes a powerful tool for data management and decision support of the radar "Kykkos". Future work will focus on fine-tuning decision-making rules and further exploiting Abacus platform for improving its competence.

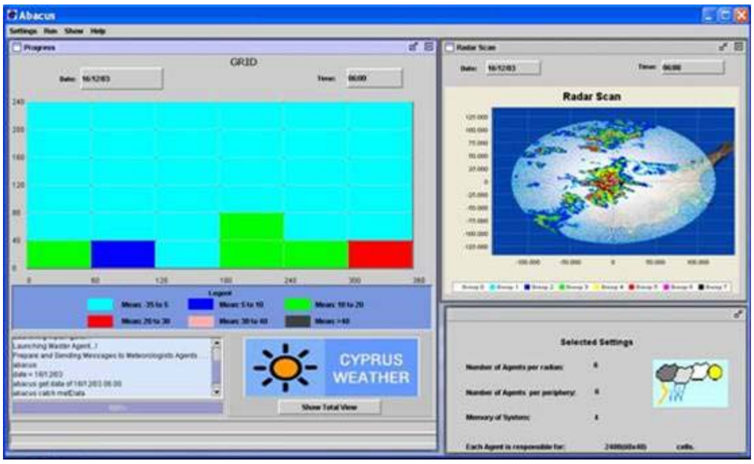


Figure 6. The main GUI of the platform, depicting the current scan (on the right) and qualitative indices of the weather conditions (on the left).

References

Athanasiadis, I.N., Mitkas, P.A. 2004a. An agent-based intelligent environmental monitoring system**,** Management of Environmental Quality, 15 (3), pp.238-249.

Athanasiadis, I.N., Mitkas, P.A. 2004b. Applying agent technology in environmental management systems under real-time constraints. In: C. Pahl, S. Schmidt, A. E. Rizzoli, and A. Jakeman (eds), Transactions of the 2nd Biennial Meeting of the Int'l Environmental Modelling and Software Society: "Complexity and Integrated Resources Management", iEMSs, Osnabrueck, Germany, vol.2, pp. 531-536.

Athanasiadis, I.N., Mitkas, P.A. 2004c. Software Agents for Assessing Environmental Quality: Advantages and Limitations. In: 18th Intl Conference Informatics for Environmental Protection: "Sh@ring" (EnviroInfo 2004), éditions du Tricorne, Geneva, Switzerland, vol.2, pp. 303-306.

Athanasiadis, I.N., Mitkas, P.A. 2005a. A distributed system for managing and diffusing environmental information. In: 5th Intl Exhibition and Conference on Environmental Technology (HELECO '05), Technical Chamber of Greece, Athens, Greece, pg. 213.

Athanasiadis, I.N., Mitkas, P.A. 2005b. A methodology for developing environmental management systems with software agents. Environmental Modeling and Software, (to appear), 2005.

Bellifemine, F., Poggi, A., Rimassa, G. 2001. Jade: A FIPA2000 compliant agent development environment. In: Proceedings of the 5th Intl Conference on Autonomous Agents, pp. 216–217, Montreal, Canada, ACM.

FIPA. 2002. Agent Management Specification. Doc. No. SC00023J, Foundation of Physical Intelligent Agents, Geneva, Switzerland.

Jennings, N. R., Sycara, K., Wooldridge M. J. 1998. A roadmap of agent research and development, Autonomous Agents and Multi-Agent Systems,1 (1), pp. 7-38.

Michaelides, S., Gabella, M., Papadakis, M. 2003. Radar adjusted rain fields in a complex-topography region (Cyprus) and comparison with TRMM data, Deliverable 3.2 - First Report on the Selected (Historical and Recent) Events Data Transmission. VOLTAIRE project (EVK2-2002-CT-00155),.

Milis, M. 2004. A multi-agent system for managing and controlling Meteorological Radar Data. Diploma Thesis, P. A. Mitkas and I. N. Athanasiadis (advisors). Electrical and Computer Engineering Dept, Aristotle University of Thessaloniki.

Noy, N. F., M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. IEEE Intelligent Systems, 16(2):60–71, 2001.

Wagner, G. The Agent–Object–Relationship meta-model: Towards a unified conceptual view of state and behavior. Information Systems, 28(5): 475–504, 2003.

Wooldridge, M., N. R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems, 3(3):285–312, 2000.