Environmental Modelling & Software 24 (2009) 1264–1273

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A multi-agent system for meteorological radar data management and decision support☆,☆☆

Ioannis N. Athanasiadis [a,*], Marios Milis [b], Pericles A. Mitkas [c], Silas C. Michaelides [d]

[a] *Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland*
[b] *SignalGeneriX Ltd, Cyprus*
[c] *Electrical and Computer Engineering Dept, Aristotle University of Thessaloniki, Greece*
[d] *Meteorological Service of Cyprus, Larnaca, Cyprus*

A R T I C L E I N F O

Article history:
Received 21 January 2008
Received in revised form
26 April 2009
Accepted 27 April 2009
Available online 12 June 2009

Keywords:
Agent-oriented software engineering
Environmental management and decision-support systems
Doppler radar data monitoring
Assessment and reporting
Meteorology software applications
Distributed decision support

A B S T R A C T

The continuous processing and evaluation of meteorological radar data require significant efforts by scientists, both for data processing, storage, and maintenance, and for data interpretation and visualization. To assist meteorologists and to automate a large part of these tasks, we have designed and 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. Abacus agent system identifies potentially hazardous incidents, disseminates preprocessed information over the web, and enables warning services provided via email notifications. In this paper, Abacus' agent architecture is detailed and agent communication for information diffusion is presented. Focus is also given on the customizable logical rule-bases for agent reasoning required in decision support. The platform has been tested with real-world data from the Meteorological Service of Cyprus.

1. Introduction

Environmental data management and surveillance, especially for hazard identification and forecasting, require significant efforts for the operational integration of environmental monitoring devices, such as sensor networks, radars and satellites. Information technology contribution is essential for overcoming interoperability obstacles and disseminating environmental information effectively and accurately. Existing monitoring devices and networks need to be coupled with software applications that enable open services, making recordings available to wider audiences. Issues of transparency, interoperability and reusability are at stake. To achieve these objectives, legacy environmental monitoring infrastructure require to be extended with modular, flexible software interfac es that provide with inputs to Environmental Decision-Support Systems for hazard identification and incident forecasting.

In this background, this paper presents the development of a software system, called Abacus, that effectively captures, processes and delivers data recorded by a meteorological Doppler radar. Abacus has been developed as a multi-agent software tool for meteorological radar data management and decision support. Agent-based computing has been selected as a means for achieving modularity and service-orientation properties.

Cross-disciplinary innovations related to both software agent technology and meteorology are presented in the paper. Section 2 gives the background of the meteorological radar 'Kykkos' of the Meteorological Service of Cyprus, and the common problems in its daily operation. In Section 3, Abacus system requirements are specified and the adoption of an agent-oriented approach is discussed. Next, Section 4 specifies the architectural design and details Abacus' generic agent types, each one of which undertakes specific tasks. Of particular interest is a community of artificial 'meteorologist' agents that take over decision-making responsibility for parts of the radar scan. Each member of the community is assessing the weather in its own niche. Both local and global decision making of Abacus' agents using rules is detailed as well. Section 5 presents agent communication and the ontology of the system, along with

☆ This work was first reported at ISESS 2005.
☆☆ Abacus software is available for free for academic use. Please contact authors to obtain a copy.

* Corresponding author. IDSIA, Galleria 2, CH-6928 Manno, Lugano, Switzerland.
E-mail address: ioannis@athanasiadis.info (I.N. Athanasiadis).

1364-8152/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2009.04.010

system implementation details and the demonstration, using actual recordings from 'Kykkos' radar. Finally, the paper concludes with discussing the main findings of the reported work, gives pointers to related work and raises issues for future work.

2. The meteorological radar Kykkos

2.1. Meteorological radar details

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 located. 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 predefined scan strategies and consequently are stored in a local hard disk.

2.2. Radar use and limitations

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.

These activities require the engagement of meteorological scientists, who are responsible for acquiring radar's data, preprocessing them appropriately, and ultimately making decisions. This process involves data filtering and restoration activities, as radar's reflections are disturbed by natural obstacles that cause beam's occultation and ground clutter problems (Golz et al., 2006).

Beam Occultation is observed when the radar beam is 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 latter case the radar beam can pass over the mountain, but the measurement is disturbed.

Ground Clutter is the disturbance that is caused when the radar beam hits the terrain, and an echo occurs. This echo can be considered as rain signal by mistake. 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.

Beam occultation and ground clutter phenomena depend on the surrounding terrain morphology. Fig. 1 illustrates a segment of radar reflections, where areas of zero velocity echoes due to some mountains in the region that destruct beam reflections are apparent. These disturbances have been taken into consideration through the design of the Abacus system, 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). Either operating in the surveillance, or in the off-line mode, the continuous evaluation of radar data requires significant efforts by the meteorologists, for data processing, storage, and maintenance, along with data interpretation and visualization.

Table 1
Kykkos' radar parameters.

Transmitter–receiver	
Peak power	158 kW
Frequency	5.7 GHz
PRF	250 Hz and 1180 Hz
Pulse duration	2 μs 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	500 m
Number of power levels	80 (−15 dBz to 65 dBz)

3. System requirements and methodology

3.1. System goals

For supporting meteorologists in their activities, we developed a software agent-based system for managing radar data and decision support, called *Abacus*. 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 system is envisioned as a platform of autonomous, artificial 'meteorologist agents' that undertake Kykkos radar data management and exploit appropriately all information that it produces. Specifically, the system's main objectives concern:

(a) the data review, transformation and preprocessing of radar's scans;
(b) the identification of weather conditions at real time and their evolution through time;

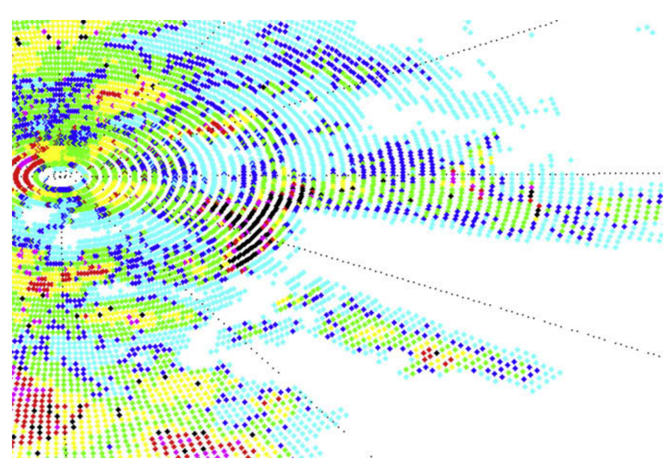

Fig. 1. Radar reflections are disturbed by natural obstacles, therefore some measurements are distracted or missing.

(c) the provision of information services to authorized personnel and the public using email notifications, distribution of alarms over the internet, etc.

In order to achieve these goals, Abacus agents are required to implement a set of functionalities. First comes the processing of successive radar scans. This activity involves:

i. the capture of reflectivity values produced by the radar, their transformation and filtering (taking into account ground clutter and beam occultation biases),
ii. the calculation of various quantitative metrics and qualitative indices within the radar's range, generating valuable secondary information,
iii. the depiction of data, metrics and indices in a variety of graphical representations, and
iv. the presentation of raw radar data, generated secondary information, and diagrams to the final users, through a friendly and functional user interface.

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

Second comes the provision of alerting services whenever predefined conditions are satisfied in certain areas of interest. Users are enabled to define rules for describing various weather conditions, based upon the extracted metrics and indices. These rules incorporate both time and space restrictions and each one describes a certain type of incident and is associated with a certain response. The latter are subsequently disseminated in three different ways:

i. by sending an email message to a predetermined list of receivers, containing location, time and the conditions associated to the event detected.
ii. by raising a sound signal that notifies the operator scientist staying close.
iii. by posting a warning message on a webpage, that contains location, time and the conditions that activated the alarm.

The above-mentioned functionalities are assigned to a community of cooperating software agents, that form synergies for delivering added-value services. The design of the agent community and the main agent types are discussed in Section 4.

3.2. Methodology adopted

An agent-oriented software engineering approach was adopted. Specifically, software agents were considered as the building blocks of the system during both the design and implementation stages. With the notion of an agent, we define a software entity characterized by autonomy, reactivity, pro-activity, and social ability. 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 cooperating agents.

There are several reasons for using software agents in both the design and the implementation of an environmental information management system, as Abacus, discussed in detail in Athanasiadis and Mitkas (2009). The main advantage foreseen from this choice is that agent-orientation implies a highly modular, service-oriented architectural design, while on the implementation side a loosely coupled integration in an open environment comes as an intrinsic characteristic of agent systems. In this way, the adoption of agent-orientation maximizes the potential of reusing and extending the system. We also note the fact that certain types of software agents have abilities to infer rationally and therefore support the decision-making process. Several agents, structured in groups, can share perceptions and operate synergistically to achieve overall goals (Jennings et al., 1998). This characteristic will be ultimately exploited in Abacus for distributed decision making.

Furthermore, developing intelligent software applications for environmental management and assessment with software agents is advantageous, as the notion of an agent is easily comprehensible by natural scientists, environmentalists, economists, social scientists and software engineers. In this way, end-users can be involved in the software design phase, and through an agent-oriented specification process, requirements can be elicited more easily (Athanasiadis, 2007). Also, value conflict problems that are too common in environmental problems can be tackled more efficiently.

Software agent technology benefits for rapid prototyping and software reusability can be also considered among the 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 tool and all agent types that constitute the system.

4. The Abacus agent-based system

4.1. Abstract architecture and end-user types

The Abacus platform is a middleware application where software agents work together for providing information services, as illustrated in Fig. 2. Raw reflection data produced by Kykkos radar are captured and processed by the platform for supporting data management, web notification, email warning and sound alarm services to the final users.

Abacus end-users are clustered in three distinct groups: (a) the radar operator, who administrates the system, (b) the meteorology service personnel, who have full access to raw data, secondary information, graphical illustrations, and alarm/notification services, (c) the indirect users (the public), who could be granted access to the system, in future extensions via web-based information services.

In its current formulation, Abacus is directed towards the professionals who can assess the risk and take specific measures for the mitigation of the hazards; it is not meant at this stage to provide direct access to the public because the interpretation of the alerts by non-professionals may lead to misleading consequences

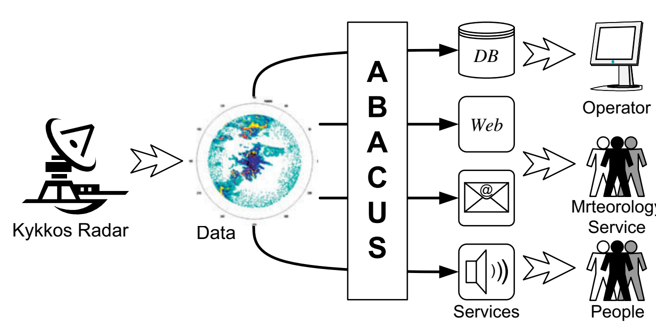

Fig. 2. Abacus abstract architecture, as a middleware for providing information services.

(at a later stage, trained individuals may take advantage of Abacus alerts, such as journalists, local administrators, farmer association representatives, television station managers, etc), though alerting meteorologists to significant thundery situations shouldn't be underestimated. Alerting systems recently built in the Meteorological Service of Cyprus for other hazards have proved to be an effective set of tools in the hands of expert weather forecasters. Of course the final word for the issuance of a warning rests with the weather forecaster who uses a number of other tools and supporting schemes before a final decision is taken. Similarly, other groups of professional users that can make effective use of Abacus alerts (e.g. the civil protection agencies) have to treat such alert systems with much caution and in consultation with other professional bodies.

4.2. System design and agent types

Abacus functionalities, specified in Section 3, have been realized through a layered agent-based architecture. In previous works, we have introduced layered architectures of cooperative software agents in environmental applications, providing services related to air quality and water management (Athanasiadis and Mitkas, 2004, 2009; Athanasiadis et al., 2005a,b; Athanasiadis, 2006). A similar approach was followed for Abacus design, where generic software agent types (Athanasiadis and Mitkas, 2009) are undertaking system's tasks. Specifically, system activities have been clustered in three functional, cooperative layers:

1. The *contribution* layer, that is responsible for acquiring radar scans and data filtering preprocessing activities.
2. The *management* and *processing* layer, where secondary information (metrics and indices) is derived, and decision rules are applied for identifying potentially interesting incidents.
3. 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 defined for realizing system functionalities. The Abacus platform architecture and the synergies between the three layers are illustrated in Fig. 3. Agent roles correspond to specific agent behaviours, which have been realized by six generic agent types. In the contribution layer resides the *radar agent*, in the management and processing layer the *meteorologist agent* and the *abacus agent*, while the distribution layer embraces the *database agent*, the *alarm agent* and the *user interface agent*.

The system goals of the various agent types and their responsibilities are as follows:

The **radar agent** retrieves radar scans and restores data biased by beam occultation and ground clutter effects. To perform this activity the radar agent incorporated the findings of previous studies of the Meteorological Service of Cyprus, which are tailored to the local situation of the specific radar (Golz et al., 2006).

Meteorologist agents 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 secondary information) within its sector and applies decision rules for assessing the weather conditions and issuing alarms at a local level.

The **abacus agent** summarizes all information extracted by the meteorologist agents. It is also responsible for issuing warnings at a global scale.

The **user interface agent** implements the graphical user interface with the operator (direct user). It is responsible for instantiating the platform at start up and for preparing and visualizing graphs and maps on the end-user terminal.

The **database agent** connects to the system database, and stores original radar data scans, cleaned data and extracted secondary information.

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

4.3. Abacus' operational cycle

To illustrate the synergies between Abacus platform agents, lets follow an operational cycle of the system. First, the *radar agent* captures a scan from the radar, and applies the filters for dealing with ground clutter and beam occultation problems. The preprocessed scan is then distributed to a community of *meteorologist agents* that exploit agent reasoning capabilities, through user-defined rules, for assessing the weather conditions in their area of responsibility. Each one of them, communicates their evaluations to the *DB agent*, which stores them persistently in a database, but more importantly to the *abacus agent*. The latter decides upon the overall weather outlook based on each one of meteorological agent estimations. Finally, *abacus* and *meteorologist agent* decisions are communicated to the *UI agent* that updates the operator's screen and, to the *alarm agent*, in case that any alarms have been identified.

Also, through the *UI agent*, the operator may communicate directly with the *meteorological agents* and the *abacus agent* for setting user-defined system parameters and decision rules. Rule specifications can be dynamically comprehended by the *meteorologist agents*, and are eventually activated *at runtime*.

4.4. Meteorologist agents' spatial positioning

Each meteorologist agent is responsible for a part of the radar scan. Although the approach followed is generic, lets focus on the Kykkos radar setting: each radar scan is made up of a reflection array of 240×360 measurements. The 240 array rows correspond to the 240 actinic steps (each one of which is 500 m long). The 360 columns correspond to the 360 cyclic sectors (each one of which is 1°) recorded in every radar rotation. Note that the 240×360 sized array of data is measured in polar coordinates, where the radar is in the center of the cycle, and therefore each reflection measurement corresponds to an area of different size. The closer to the radar, the bigger the resolution of the measurements.

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.[1] Each agent is identified by its polar coordinates. In Fig. 4(a), an example of a hundred agents (distributed 10×10) on the plane is illustrated: 10 agents per annulus and per sector. Each agent's area of responsibility corresponds to an annular sector and is composed of a matrix of 24×36 values. The outer annulus is covered by ten agents, and the eleventh agent is 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. Fig. 4(b) shows the information matrix for the first agent, which is an array of 24×36 reflection values. Such a homogeneous distribution of agents over the radar plane was adopted in this work,

[1] The meteorologist agents are considered to be virtually distributed over the radar plane, as marbles of an abacus, hence comes the name of the system.

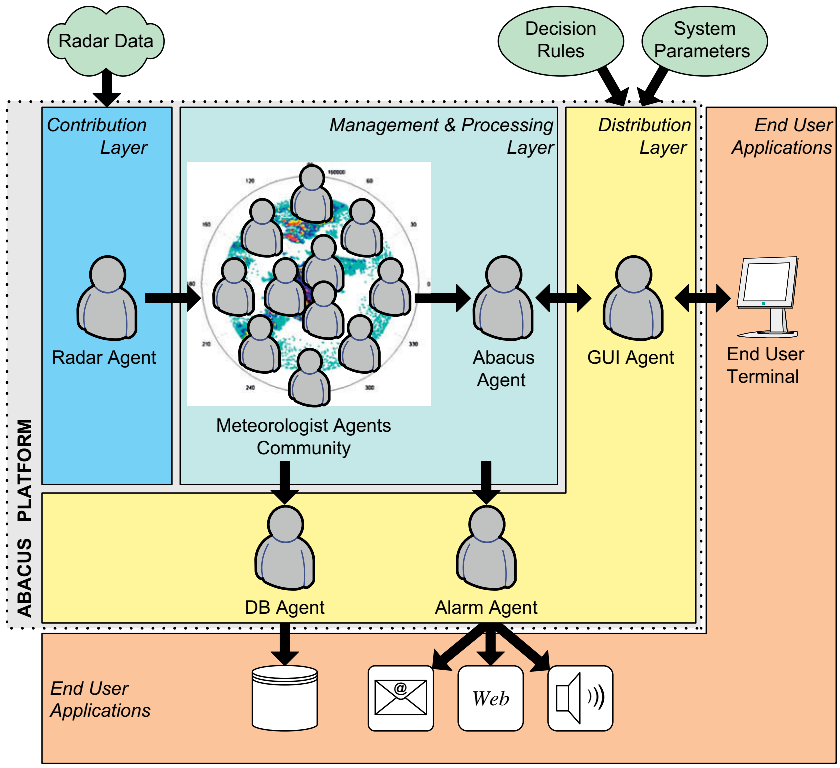

Fig. 3. Abacus platform architecture.

with the total number of agents, the number of cyclic sectors, and actinic steps, being parameters of the system. Future research may focus on studying other types of agent spatial distributions over a radar plane.

4.5. Meteorologist agents, local and global decision making

On-line, dynamic decision support is a key feature of the tool that is enabled through agent reasoning and communication. Each one of the *meteorologist agents* incorporates its own rule engine for assessing weather conditions. The rules provide with a set of weather conditions that correspond to interesting events (i.e. cloud formation, precipitation phenomena, etc) within each *meteorologist agent's* area of responsibility. All decision rules are defined through the platform GUI by the meteorological scientists at the Meteorological Service of Cyprus.

Each *meteorologist agent* rule consists out of three parts:

(a) An *assumption antecedent* that relates meteorological indices (secondary information) with certain value ranges.
(b) A *time constraint* that defines a time interval within which the assumption antecedent should be satisfied.
(c) The corresponding alarm, which is the *consequence* (decision) to which the agent concludes 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 rules' assumptions. Each *meteorologist agent* derives four indices from radar reflections that it is responsible for within an annular sector of the radar scan. They are:

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.

These indices are used for building a rule's assumption. A simple rule that could be constructed and embedded into a meteorologist agent is the following:

```
IF percentage of cells WITHIN RANGE 50–60 dBz
IS GREATER THAN 60% FOR 3 succeeding scans
THEN raise a sound alarm
```

In this example we employ the fourth index (percentage of cells), a cloud type (range 50–60 dBz), an assumption antecedent (>60%), and a time constraint (for 3 succeeding scans), for ultimately raising a sound warning.

The parameterization of the decision rules and the selection of the indices were based on an in-depth study on radar data use and radar characteristics and analysis (Gabella et al., 2006). Also, the cloud type classification is based on prior empirical experience. The definition of the rules incorporated by the agents is not the main issue of this paper, rather it is left to the scientific personnel of the Meteorological Service of Cyprus. The contribution of the approach presented is the modularisation of complex rules, their management and the activation at runtime. 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 ruleset is embedded in each agent for supporting the decision-making process of the system. A potential event can activate more than one rule, and result into several warnings that are addressed to different audiences. In this respect, each scientist involved in the process can customize his/her own ruleset and evaluate them through Abacus. With the presented approach, end-users are enabled to define cloud classifications and rules based on four indices and embed them into agents dynamically.

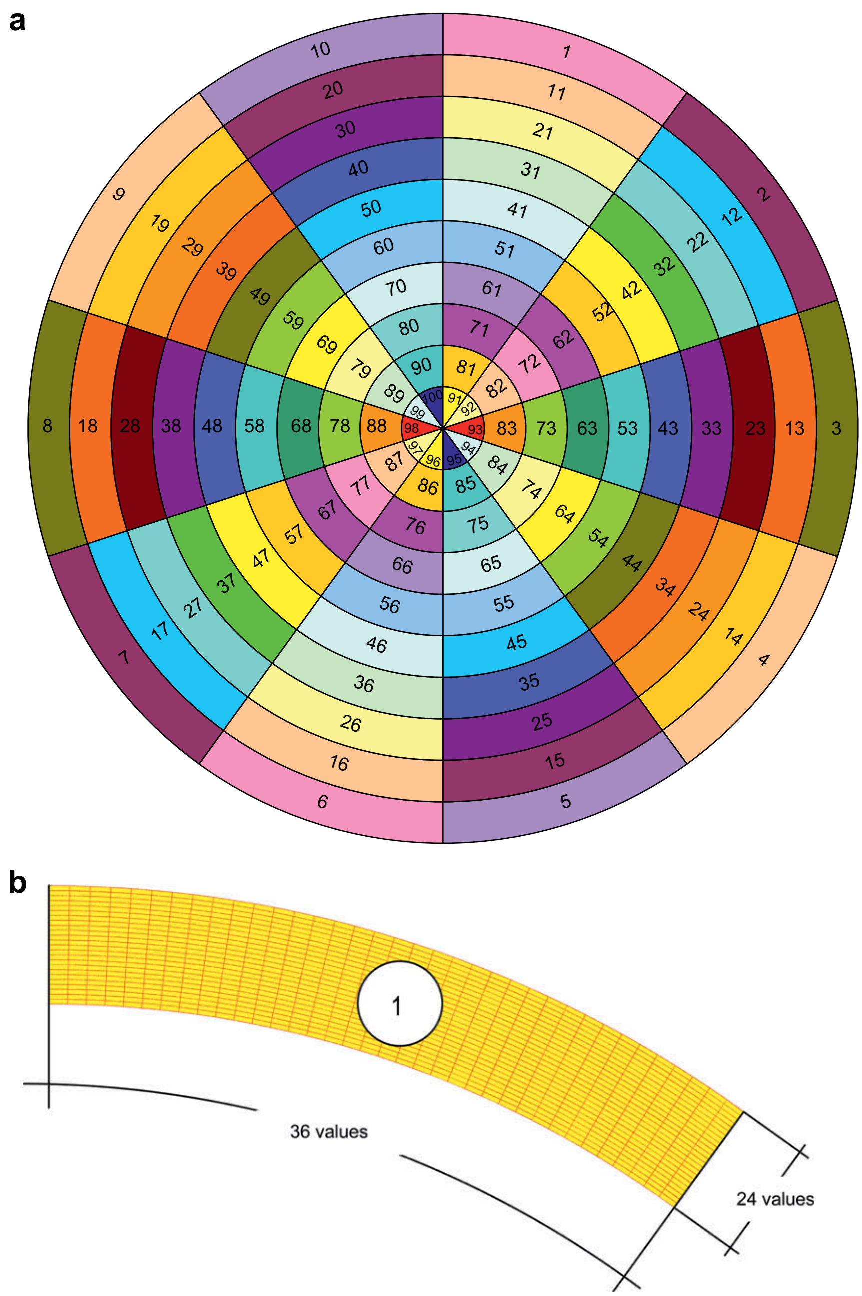

Fig. 4. An example of having a hundred agents on the radar's plane (distributed 10 × 10).

5. Agent communication, ontology and system implementation

5.1. Agent communication and Abacus ontology

Abacus agent model was specified using the GAIA methodology (Wooldridge et al., 2000; Zambonelli et al., 2003). The software agent interaction has been specified using the Agent–Object Relationship Modeling Language (AORML), introduced by Wagner (2003). Fig. 5 illustrates the AORML external agent diagram, which defines all communication among Abacus agents and users. In this diagram you may notice that both end-user and artificial agents are depicted, and the communication acts among them are specified. During these developments we realized that it was much easier to communicate the software specifications through agent-oriented diagrams, as software entities, similar to the actual users, have human-like characteristics, e.g. 'undertake tasks', 'are responsible for assessing the weather conditions', 'communicate to each other' and 'can update their decision-making strategies'.

Agent communication is a key feature in Abacus platform that enables both the propagation of information and the provision of decision-support services. Analytically, Abacus multi-agent community operates in eleven steps, shown Fig. 5 and detailed below:

Step 1. The Application User defines the system settings and selects the decision rules.
Step 2. *UI agent* initiates all platform agents at runtime, based on user settings.
Step 3. *Radar agent* reads radar data and filters/preprocesses them.
Step 4. Filtered radar scan is sent to the *meteorologist agents*.

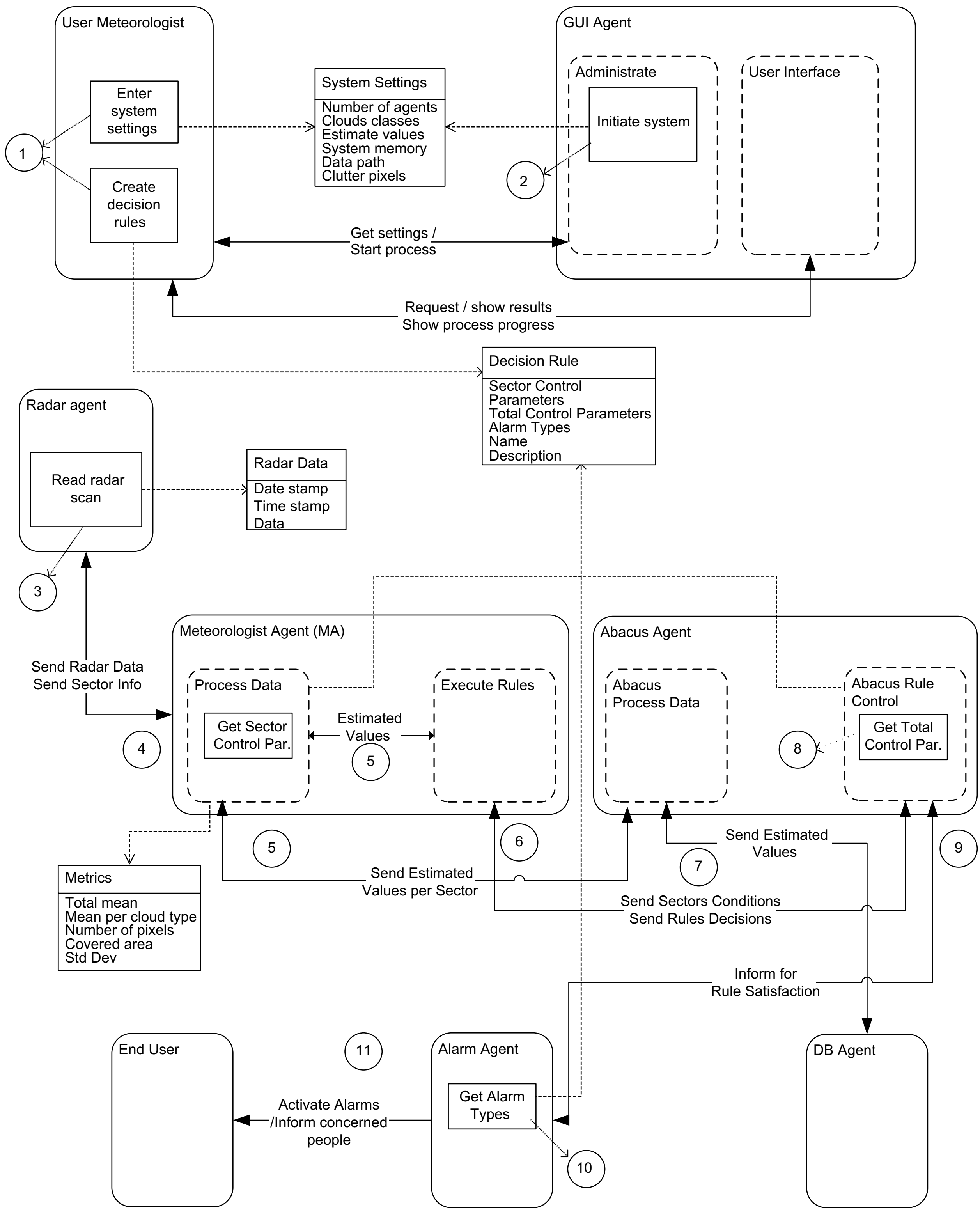

Fig. 5. Abacus' external agent diagram with AORML. Numbers correspond to steps explained in Section 5.1.

Step 5. *Meteorologist agents* calculate indices and metrics for their area of responsibility and fire their decision-making rules.

Step 6. All *meteorologist agents* send the extracted secondary information and alarms to the *Abacus agent*.

Step 7. *Abacus agent* concatenates data from all annular sectors and creates a joint 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. *Abacus agent* forwards all raised alarms to the *Alarm agent*.

Step 10. *Alarm agent* processes the raised alarms and synthesizes webpage or email content.

Step 11. *Alarm agent* activates the alarms, via web, email, or sound signal.

Information communicated by platform agents is structured. Agent messages follow a generic ontology developed with the Protégé-2000 ontology editor (Noy et al., 2001). Part of the Abacus Ontology developed is presented in Fig. 6, where the concepts of the system, along with agent actions and predicates are specified. The slots (properties) of the various concepts have been configured in order to contain the appropriate content communicated by the agents.

5.2. Implementation and demonstration

The Abacus platform was implemented in Java. JADE suite has been used for agent development (Bellifemine et al., 2003). Agent design and implementation in Abacus conforms to the FIPA specifications (FIPA, 1999–2002) for agent communication and message routing. This means that other agent systems could be registered in the Abacus agent platform and communicate with them. In such a way system functionalities and services could be extended. Note that the Abacus user is required neither to have any programming skills, nor to understand the internal functionalities of the platform, as system parameterization and rule management is done through the GUI. The advantages of the implemented system are its user-friendly interface and its open, easy-to-parameterize implementation.

The Abacus platform has been demonstrated to the Meteorological Service of Cyprus. Abacus has been evaluated on an off-line mode (while the surveillance mode is also enabled) and may handle the actual radar data scans at operational time frame (1–2 min per scan). The software validation of the system has been done using recordings from past events. Prior studies results for the detection of ground clutter and beam occultation phenomena of Kykkos' radar (Golz et al., 2006) have been used by the system to filter raw radar scans. Experts from the Meteorological Service of Cyprus have evaluated Abacus by defining their own rules, embedded them into the agents and let Abacus evaluate radar scans that have been previously recorded, and fire alarms.

As an example of the system operation, two screenshots of the operator's GUI are shown in Fig. 7. Through the main GUI the user can review a graphical representation of the actual radar scan (on the top right) along with a colour-based overview of agents' decisions (on the left). The first screenshot (Fig. 7(a)) corresponds to a sunny day, while the second one is for a stormy day (Fig. 7(b)). A single view on the main GUI can provide the operator a full view of the situation monitored by the system. Note that by double clicking on the agent grid on the left the user may access graphical representations of the metrics and indices calculated by each *meteorologist agent*.

Experience within an operational weather forecasting office has shown that any new development, whether this is a new forecasting methodology/practice or the updated version of an existing one, requires systematic training of the personnel and the proper introduction into the routine operations. Normally, the management of the above change in practice includes a period of verification and evaluation that is essential to enhance the confidence of the operators and users but also to support the superiority and advantages over existing practices. This can take a long period of time since the required output is the homogeneous and standardized utilization of the new practice.

Nevertheless, a first evaluation of the Abacus software has shown that it has a great practical application within an operational weather forecasting office. The criteria used are those which are conceptually used by weather forecasters. In other words, the rules under which Abacus recognizes important events are as good as those of a human operator. In addition, the system has an advantage over the human operator because it relieves the meteorological staff from the obligation to attend the radar output continuously. In this respect, Abacus can free the human operator from a demanding task and he/she can devote more time in other operations. Also, Abacus can objectively attend the radar data limiting also the unavoidable fatigue of the operating staff.

6. Discussion, related work and conclusions

This paper presented Abacus, that constitutes a powerful tool for data management and decision support for the radar 'Kykkos'. The main competencies of the system include both information-processing and decision-making services, supplied in an open, distributed environment of cooperating software agents. The system presented has demonstrated how software agent technology can be utilized for providing advanced services in meteorology, and for 'opening up' legacy monitoring devices, such as a meteorological Doppler radar. Furthermore, a distributed method

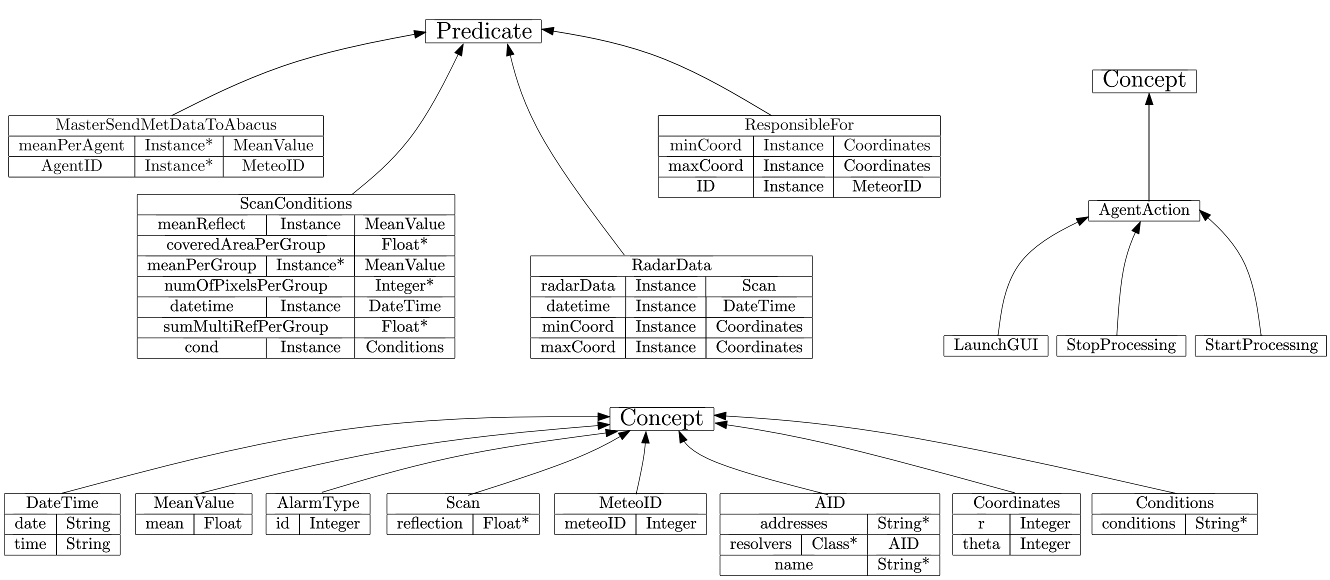

Fig. 6. Abacus ontology: concepts, predicates and agent actions.

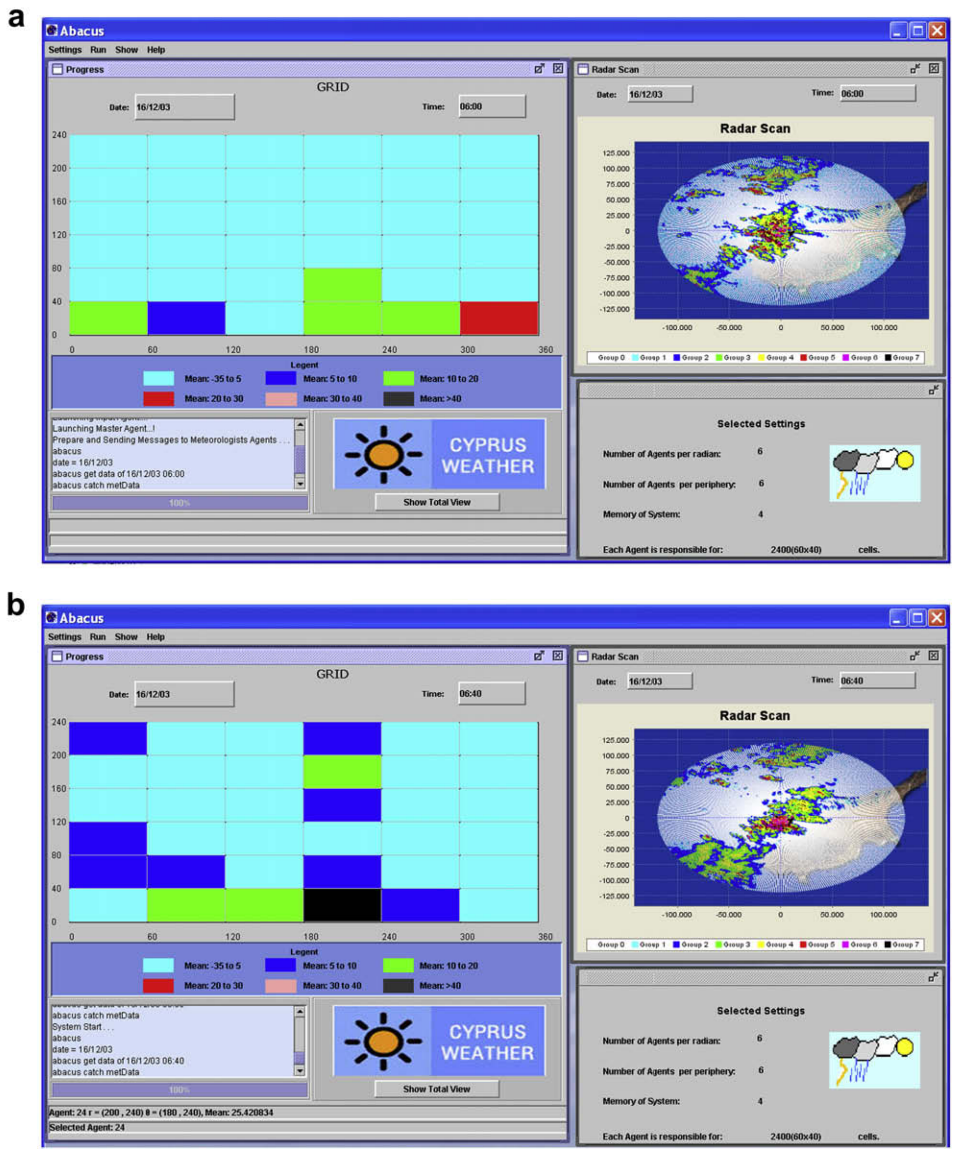

Fig. 7. The main GUI of the platform, depicting the current scan (on the right) and qualitative indices of the weather conditions (on the left).

for extracting secondary information from raw radar beam reflection data, as quantitative metrics and qualitative indices has been presented. For the system design, a three-layer architecture of cooperating services, provided eventually by agents, has been employed. System specification using agent-based techniques has improved the trust of the end-users to the system, as they have a deeper understanding of its functionality. Finally, the system has been evaluated with a set of recordings from the radar, that correspond to different weather conditions.

Agent technologies have been welcomed in ecological and environmental applications mainly as a metaphor for decomposing complex systems and studying the emergence of collective behavior (Olson and Sequeira, 1995). Agent-based techniques have been used for modeling several environmental fields, including socio-ecological systems (Batten, 2007), human–wildlife interactions (Anwar et al., 2007), integrated water resources management (Barthel et al., 2008), residential power consumption (Xu et al., 2008), epidemiology (Rao et al., 2009), water trading (Smajgl et al., 2009), land-use change (Bithell and Brasington, 2009), environmental health (Sokolova and Fernández-Caballero, 2009). An interesting collection of advanced agent-based environmental management systems has been recently compiled by Cortés and Poch (2009). However, agent technology has been adopted as a tool for software design and implementation of environmental applications only in a limited, rather fragmented way, as Athanasiadis (2005) pointed out after studying twenty-three systems that utilize agent technology at different stages of environmental system development. Abacus demonstrated that agent technologies can be applied at all stages of environmental software lifecycle, from conception to deployment.

Also, there is limited prior experience with agent-based computing in meteorology applications. The Australian Bureau of Meteorology has reported the use of agent technologies (Dance and Gorman, 2002; Dance et al., 2003; Mathieson et al., 2004) for monitoring in real time the current terminal area forecasts (forecasts for areas around airports) and alerting forecasters. Also, agent technology for decision support from radar data has been exploited by Hughes and Lewis (2005), where a layered agent architecture has been employed for detecting low observable targets. To our knowledge the work presented here is the first one that employs a software agent approach for managing and enriching meteorological radar data. Andersson and Rönnbom (2001) have discussed the use of agents with radar data but they did not specifically apply them to weather information. Dance and Potts (2002) reported using agent technology with Doppler weather radar data in an attempt to locate the presence of microbursts. To the best of our knowledge this is the first work that employs a software agent approach for managing and enriching reflectivity radar data for managing the evolution of cloud patterns and especially those which can lead to hazardous events.

Future work will focus on fine-tuning decision-making rules and further exploiting Abacus platform for improving its functionality, i.e. by adding advanced dissemination services, and making them available to wider audiences. Areas for future research may include the non-symmetric positioning of meteorologist agents. For example, certain regions where human-related activities are held (i.e. ports and airports, major cities, ship routes, agricultural fields, etc) might be of particular interest. Another open issue is related to the *meteorologist agent* self-adaptation and learning through time-evolving conditions. Also, alternative system settings may consider agents that are associated to each cloud formulation, which ultimately are moving within the area monitored by the radar. In a future system we could have agents that monitor clouds and have them moving over the radar plain. Finally, a fully integrated meteorological information system is envisioned, where data recorded from diverse sources as ground stations, radar reflections and satellite images will be integrated for improving the supplied services. Such a system will eventually need an open, modular architecture, analogous to the one presented in this paper, for deploying synergies among various primary data sources (ground stations, radar, satellite).

Acknowledgements

The authors wish to thank the three anonymous reviewers for their constructive comments.

References

Andersson, L., Rönnbom, A., 2001. Intelligent agents – a new technology for future distributed sensor systems? Master Thesis, Department of Informatics, School of Economics and Commercial Law, Göteborg University.

Anwar, S.M., Jeanneret, C.A., Parrott, L., Marceau, D.J., 2007. Conceptualization and implementation of a multi-agent model to simulate whale-watching tours in the St. Lawrence Estuary in Quebec, Canada. Environmental Modelling and Software 22 (12), 1775–1787.

Athanasiadis, I.N., 2005. A review of agent-based systems applied in environmental informatics. In: Zerger, A., Argent, R.M. (Eds.), MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, Melbourne, Australia, pp. 1574–1580.

Athanasiadis, I.N., 2006. An intelligent service layer upgrades environmental information management. IT Professional 8 (3), 34–39.

Athanasiadis, I.N., 2007. Towards a virtual enterprise architecture for the environmental sector. In: Protogeros, N. (Ed.), Agent and Web Service Technologies in Virtual Enterprises. Information Science Reference, Hershey, PA, USA, pp. 256–266.

Athanasiadis, I.N., Mentes, A.K., Mitkas, P.A., Mylopoulos, Y.A., 2005a. A hybrid agent-based model for estimating residential water demand. Simulation: Transactions of the Society for Modeling and Simulation International 81 (3), 175–187.

Athanasiadis, I.N., Solsbach, A., Mitkas, P.A., Marx-Gómez, J., 2005b. An agent-based middleware for environmental information management. In: Leal-Filho, W. (Ed.), 2nd International Symposium on Information Technologies in Environmental Engineering (ITEE-2005). Shaker Verlag, Magdeburg, Germany, pp. 253–267.

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

Athanasiadis, I.N., Mitkas, P.A., 2009. A methodology for developing environmental information systems with software agents. In: Cortés, U., Poch, M. (Eds.), Advanced Agent-Based Environmental Management Systems. Whitestein Series in Software Agent Technologies and Autonomic Computing, vol. 21. Birkhäuser, pp. 119–138.

Barthel, R., Janisch, S., Schwarz, N., Trifkovic, A., Nickel, D., Schulz, C., Mauser, W., 2008. An integrated modelling framework for simulating regional-scale actor responses to global change in the water domain. Environmental Modelling and Software 23 (9), 1095–1121.

Batten, D., 2007. Are some human ecosystems self-defeating? Environmental Modelling and Software 22, 649–655.

Bellifemine, F., Caire, G., Poggi, A., Rimassa, G., September 2003. JADE – a white paper. EXP in Search of Innovation 3 (3), 6–19.

Bithell, M., Brasington, J., 2009. Coupling agent-based models of subsistence farming with individual-based forest models and dynamic models of water distribution. Environmental Modelling and Software 24 (2), 173–190.

Cortés, U., Poch, M. (Eds.), 2009. Advanced Agent-Based Environmental Management Systems. Whitestein Series in Software Agent Technologies and Autonomic Computing, vol. 21. Birkhäuser.

Dance, S., Gorman, M., July 2002. Intelligent agents in the Australian Bureau of Meteorology. In: Proceedings of the 1st International Workshop on "Challenges in Open Agent Systems" held at AAMAS'02. Bologna, Italy.

Dance, S., Gorman, M., Padgham, L., Winikoff, M., 2003. An evolving multi agent system for meteorological alerts. In: Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS-03. ACM Press, pp. 966–967.

Dance, S., Potts, R., 2002. Microburst detection using agent networks. Journal of Atmospheric and Oceanic Technology 19 (5), 646–653.

FIPA, 1999–2002. Agent Specifications. Technical Report. available online:. Foundation for Intelligent Physical Agents http://www.fipa.org.

Gabella, M., Joss, J., Perona, G., Michaelides, S., Jan. 2006. Range adjustment for ground-based radar, derived with the spaceborne TRMM precipitation radar. IEEE Transactions on Geoscience and Remote Sensing 44 (1), 126–133.

Golz, C., Einfalt, T., Michaelides, S., 2006. Quality control of rainfall measurements in Cyprus. Meteorological Applications 13, 197–201.

Hughes, E., Lewis, M., Feb. March 2005. Intelligent agents for radar systems. Electronics Systems and Software 3 (1), 39–43.

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

Mathieson, I., Dance, S., Padgham, L., Gorman, M., Winikoff, M., 2004. An open meteorological alerting system: issues and solutions. In: ACSC '04: Proceedings of the 27th Australasian Conference on Computer Science. Australian Computer Society, Inc., Darlinghurst, Australia, pp. 351–358.

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

Olson, R.L., Sequeira, R.A., 1995. An emergent computational approach to the study of ecosystem dynamics. Ecological Modelling 79, 95–120.

Rao, D.M., Chernyakhovsky, A., Rao, V., 2009. Modeling and analysis of global epidemiology of avian influenza. Environmental Modelling and Software 24 (1), 124–134.

Smajgl, A., Heckbert, S., Ward, J., Straton, A., 2009. Simulating impacts of water trading in an institutional perspective. Environmental Modelling and Software 24 (2), 191–201.

Sokolova, M.V., Fernández-Caballero, A., 2009. Modeling and implementing an agent-based environmental health impact decision support system. Expert Systems with Applications 36 (2), 2603–2614.

Wagner, G., 2003. The agent–object-relationship metamodel: towards a unified conceptual view of state and behavior. Information Systems 28 (5), 475–504.

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

Xu, M., Hu, Z., Wu, J., Zhou, Y., 2008. A hybrid society model for simulating residential electricity consumption. International Journal of Electrical Power & Energy Systems 30 (10), 569–574.

Zambonelli, F., Jennings, N.R., Wooldridge, M., 2003. Developing multiagent systems: the GAIA Methodology. ACM Transactions on Software Engineering and Methodology 12 (3), 317–370.