

Development of a Multi-Agent-Based Simulation System for Air Quality Analysis

Elia Georgiana DRAGOMIR

Petroleum-Gas University of Ploiești,
39, București Blvd, Ploiesti, 100680, Romania,
elia.dragomir@yahoo.com

Abstract: The implementation of distributed collaborative intelligence can improve the efficiency of agent-based environmental systems (for environmental networks) in the context of solving emergency situations in real time, such as the critical situation of severe air pollution with an important impact in a given region. In this paper it is presented a distributed multi-agent system for air quality monitoring and analysis (AQDMAS) whose first version was implemented in Zeus, a Java-based toolkit for intelligent agents' development. The study is focused on the possibilities of the computer network distribution of the software agents. The purpose of the system is to verify the air quality standard fulfilment, and if, in case some exceedances occur, to inform the supervisor in order to take a decision. For this version there were designed seven intelligent agents placed in a local computer network in the Petroleum-Gas University of Ploiesti campus in Romania. The results revealed the effectiveness of the proposed monitoring and analyzing technique and the possibility of autonomous built-in simulation of tracking the air quality evolution with distributed resources.

Keywords: distributed multi agent system, air quality index, environment pollution, BDI logic.

1. Introduction

Air quality monitoring and analysis has become a very important environmental management activity in all European countries. Therefore most countries developed or have under development a national network of sites that monitor the concentrations of various air pollutants, depending on the local air pollution sources. In Romania, there is a National Network for Air Quality Monitoring that provides public measurements of some air pollutants in different locations. Thus a network is geographically distributed and can be modelled as a multi-agent system, each local station being viewed as an intelligent agent.

A brief review of the literature makes the evidence of the intensive research work already done in this domain. The monitoring and analysis of the environment problems is a domain where agent-based technology can provide proper solutions. Some research projects had reported in the literature the progress made in this direction [6], [12]. The potential use of multi-agent systems in real time environmental monitoring, analysis, forecasting, and control raises new challenges for the researchers working in the area of distributed intelligent systems [22]. Other applications of multi-agent systems in the environmental sciences have been reported in the literature: a self-organizing multi-agent system for simulating the processes responsible for the distribution of water

availability over space and time in a semi-arid river basin [21], or for real time control of water distribution networks [3], and a multi-agent system for drinking water quality analysis from a distribution network [13].

Among these, the implementation of distributed collaborative intelligence can improve the efficiency of agent-based environmental systems (for environmental networks) in the context of solving emergency situations in real time, such as, for example, the critical situation of a severe air pollution with an important impact in a given region, amplified by specific meteorological conditions. An example is a hybrid multi agent system based on a reinforcement learning approach for real time air pollution monitoring presented in [16]. The design of a two – level hierarchical multi agent system for the vehicle emission monitoring is modelled in Khoo and Meng [8]. In [24] is presented a conceptual design of a multi agent wireless sensor actuator system for the indoor air pollution. In Roche et al [20] is designed a multi-agent gas turbine power plant systems and Chappin and Dijkema proposed an agent based model for the electricity production systems in order to explore the impact of CO₂ emission-trading [1].

Our previous works in this research area include the references: Oprea et al. [14], a multi-agent system for dam monitoring, the AirQMAS presented in Oprea et al. [15] and the AgentAirPol System in Petre [17].

This paper presents a multi-agent based model of an intelligent system for air quality monitoring and analysis. As a case study a particular first version of the system it is implemented in Zeus, a Java-based toolkit for intelligent agents' development [2]. The preliminary experiments presented here are focused on the possibilities of the computer network distribution of the software agents.

One of the main objectives of this multi-agent system is to develop a BDI logic based model for a multi-agent system applied in the air quality monitoring domain. The model is design in order to reflect the geographical distributed characteristic, the real time system response and a high level of flexibility over the entire life-cycle: the system model is designed to adapt to most changes in the structure of the network, whether they are intentional (e.g., the addition of a sensor or stopping one for maintenance) or not (e.g., after an outage). Multi-agent systems (MASs) enable this flexibility due to their inherent characteristics.

The rest of the paper is divided as follows. Section 2 briefly presents the MAS area of research. Section 3 describes the design of the agent's model, including its conceptual framework, implementation process and knowledge evaluation aspects. Section 4 presents a discussion based on the experimental results, and Section 5 concludes the paper and identifies directions for future research.

2. Agents and MAS

MAS is a collection of autonomous entities called agents, which interact with each other and their environment in a cooperative or competitive way to achieve individual or group objectives. Contrary to traditional modelling techniques, MAS is not expressed in terms of variables, functions and equations, but in terms of agents, objects and environments [22]. According to Wooldridge [23], the primary advantages of MAS are decentralized control, robustness, simple extensibility, shared expertise and resources.

An agent is a computing entity that possesses the following four features: reactivity, autonomy, interaction and initiative [23]. Autonomy is the main characteristic of an agent; in other words, agents are capable of acting independently and controlling their internal states, as opposed to traditional event-

driven approaches. Weiss [22] defines a multi agent reactive system as a system that maintains continuous interaction with its environment and answers the changes that occur in that environment.

Thus, these agents are situated in an environment where they can interact through perception (e.g., using sensors) and action (e.g., using actuators). Each agent has a representation of the rest of the environment and takes decisions based on it. By interacting with each other, for example through message exchanges, agents can pursue their own local goals as well as the global goals of the MAS. These interactions can be set in a cooperative or competitive way, i.e., these goals can be conflicting or not. Compared to a client-server centralized system, the advantages of MAS include distribution of processing, support for a more flexible peer-to-peer (P2P) model, decentralization of control, reduction of network bandwidth use and scalability [12].

Applying the concept of distribution to build the MAS architecture has several advantages: flexibility, adaptability to changes etc. Creating a single monolithic system would require listing and taking into account all the events the system could possibly have to face [18]. By defining several agent types, the functioning of the system can be divided into several complementary pieces which results in a much simpler development. Therefore, agents can be added or deleted as the MAS runs at any time.

It also enables specifying several aspects that would not be considered in an implementation with a simulation software, e.g., communication, data transfer, system architecture, etc. [20]. The MAS may be easily deployed on the real system, as each part of the software is already well defined and testable. Moreover, this enables the MAS to be tested either by running all agents on a single computer, or by running one agent per computer. Finally, the ability to define which agent has access to which information facilitates taking into account information security aspects.

In summary, a multi agent system is suitable for complex applications that require distributed problem solving.

3. AQDMAS

3.1 System architecture

The AQDMAS (Air Quality Distributed Multi Agent System) is designed to monitor and analyze the concentrations of the air pollutants and meteorological parameters in a certain area. The system is modelling the distributed characteristics of air quality monitoring stations network. After the local air parameters values are measured, databases are used to store the recorded data. Based on the measurements of the air pollutants concentrations, local assessments are made by comparing them with the national air quality standard that provides the maximum admissible concentrations. The local air quality index values are sent to the centre so that the region air quality index may be calculated. The purpose of the system is to verify the air quality standard fulfilment and in case some exceedances occur, to inform the coordinator in order to take a decision (e.g. some control measures).

The selected MAS architecture consists of several agents, divided into three functional levels:

- 1) *Strategic* – this level includes the coordinator agent because it has a global view of the system, with access on the work teams and the pending jobs. It is responsible for distributing tasks among the teams using FIPA Contract Net Protocol. It is also enabled for receiving updates on the current state of the system from the GUI interface and also forwards the commands issued by the operators to the local supervisor agents.
- 2) *Local management* – this level includes the local supervisors of the teams. They are responsible for the communication with the Coordinator agent and with the local agents: after receiving the request from the strategic level, the local supervisor asks the local agents to send the required data (e.g. air pollutants concentration values, meteorological status etc.).
- 3) *Operational* – this level includes the local specialized agents. They measure the atmospheric parameters that are to be analyzed.

3.2 AQDMAS BDI logic based model

A multi-agent based system may be modelled using modal logics which were first published in the 50s by Kripke, Kanger, and Hintikka [7]. All these logics have similar semantics, called possible world. The bases of these modals revolve around a box (which can be read as “necessarily”, “it is known that”, “always in the future”) and a diamond operator (“it is possible”, “in some cases”) [7].

A Kripke model is a tuple $\langle S, R, \pi \rangle$ where S is a set of possible worlds, R is a accessibility relation on S , and π a truth assignment function. The general clause of the box operator for the model M and a world s is given in the equation (1) and the clause for the diamond operator in equation (2).

$$M, s \models \Box \varphi \Leftrightarrow M, s \models \varphi \text{ for all } t \text{ with } R(s, t) \quad (1)$$

$$M, s \models \Diamond \varphi \Leftrightarrow M, s \models \varphi \text{ for some } t \text{ with } R(s, t) \quad (2)$$

In the intelligent agents domain there has been developed different types of modal logics. One of these is the BDI logic, based on Kripke semantics, which describes the beliefs, desires, and intentions of agents, inspired by the work of the philosopher Bratman [7]. The AQDMAS model is designed based on the language, axioms and formulae proposed by [5], [7], and [9].

Definition 1. The language L is a tuple $\mathcal{L} = (\mathcal{P}, \mathcal{A})$ with the property $\mathcal{P} \cap \mathcal{A} = \emptyset$ where \mathcal{P} is a set of propositional symbols and \mathcal{A} is a finite set of agents.

Definition 2. Based on the language \mathcal{L} there can be inductively define a set of formulae as follows.

F1: each proposition $p \in \mathcal{P}$ is a formula.

F2: if φ and ψ are formulae then $\neg\varphi$ and $\varphi \wedge \psi$ are formulae.

F3: if φ is a formula, $i \in \mathcal{A}$, $G \subseteq \mathcal{A}$ then the following modalities are formulae: $BEL(i, \varphi)$, $E-BEL_G(\varphi)$, $C-BEL_G(\varphi)$, $GOAL(i, \varphi)$, $INT(i, \varphi)$, $E-INT_G(\varphi)$, $M-INT_G(\varphi)$, $C-INT_G(\varphi)$, $done(\varphi)$, $failed(\varphi)$, $succeeded(\varphi)$.

The formulae and their significations are described in Table 1.

Table 1. The significations of the formulae

Formula	Intended meanings
BEL(i, φ)	Agent i believes that φ
E-BEL _G (φ)	Group G of agents has a general believe that φ
C-BEL _G (φ)	Group G of agents has a common believe that φ
KNOW(i, φ)	Agent i knows that φ
E-KNOW _G (φ)	Group G of agents has a general knowledge that φ
C-KNOW _G (φ)	Group G of agents has a common knowledge that φ
GOAL(i, φ)	Agent i has the goal φ to achieve
INT(i, φ)	Agent i has the intention to achieve φ
E-INT _G (φ)	Every agent of group G has the individual intention to achieve φ
M-INT _G (φ)	Group G of agents has the mutual intention to achieve φ
C-INT _G (φ)	Group G of agents has the common intention to achieve φ

Definition 3. The Kripke model is a tuple

$$\mathcal{M} = (W, \{B_i; i \in \mathcal{A}\}, \{G_i; i \in \mathcal{A}\}, \{I_i; i \in \mathcal{A}\}, Res, \{R_i; i \in Res\} Val)$$

where

1. W – a set of possible worlds;
2. Res – a set of possible resources;
3. For all $i \in \mathcal{A}$ it holds $B_i, G_i, I_i \subset W \times W$ (the accessibility relations of each agent with respect to beliefs, goals, intentions).
 $(s, t) \in B_i$ ($sBit$) means that t is an alternative for agent i in state s related to its' believes. Similar it can be define $(s, t) \in G_i$, and $(s, t) \in I_i$.
4. For all $i \in \mathcal{A}$ it holds $R_i \subset W$, the accessibility relation for every agent related to resources.
5. $Val: (\mathcal{P} \times W) \rightarrow \{0,1\}$ is a truth assignment function.

In order to model the AQDMAS system there have been used the actions formulae briefly described in table 2.

Table 2. Actions formulae used by agents

Action	Description
create(general_report_from(SM_1, SM_2, \dots, SM_n))	Create the general report using the local reports from monitoring stations SM_1, SM_2, \dots, SM_n
create(local_report;_from($ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$))	Create the local report from monitoring station i using the measured data $ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$
store(database;($date, hour, ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$))	Store the measured data $ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$ in the specialized database from monitoring station i
measure(parameter;(SM_i))	Measure the value of atmospheric parameter j at monitoring station SM_i $i = \overline{1, n}, j = \overline{1, m}$
send(parameter ₁ (SM_i, ALS_i),	Send the parameter 1 value from station SM_i to ALS_i
store(database;(date, hour, $ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$))	Store in the local database I the measured values $ms^i_1, ms^i_2, ms^i_3, \dots, ms^i_m$
initiate_request(resources_for(i, r))	Agent i initiates a request for the needed resources to create r
get(r)	Get the resource r
is_working(Tr)	Check if the transducer Tr is working

3.3 The BDI model of the AQDMAS action plans

To fulfill the system goal, received from user interface, the coordinator agent (AC) needs to create a general report with the air quality index status. In equation (3) it is set the AC goal.

$$GOAL(AC, \text{create(general_report_from}(SM_1, SM_2, \dots, SM_n))) \quad (3)$$

$$\neg \text{can}(AC, \text{create(general_report_from}(SM_1, SM_2, \dots, SM_n))) \quad (4)$$

Equation (4) reflects the fact that AC cannot meet the system purpose on its own; therefore the communication with other agents is initiated to find out their abilities, willingness and commitment strategies. A first group G_R of agents is formed which includes along with AC the local supervisor agents ALSs.

$$G_R = \{AC, ALS_1, ALS_2, \dots, ALS_n\}$$

The common intention of the group is establish according with equation (5).

$$C\text{-INT}_{GR}(\text{create}(\text{general_report_from} (SM_1, SM_2, \dots, SM_n)) \quad (5)$$

An accurate action plan to accomplish this goal needs to be divided in subplans, subgoals and actions which are properly assigned to agents or groups of agents. The general plan P of the system has the scope φ . There are n subplans of $P = (P_1, P_2, \dots, P_n)$ each one for a monitoring station and n subgoals for $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_n)$.

3.4 The <GRD> plan

The general report design (GRD) plan can be divided in two main steps: the local reports design (SM_1, SM_2, \dots, SM_n – the local station report) and the general report design. Equation (6) presents the structure of a division SM_i : all the measured values from monitoring station i .

$$SM_i = (ms_1^i, ms_2^i, ms_3^i, \dots, ms_m^i) \quad i = \overline{1, n} \quad (6)$$

where n-number of monitoring station; ms_1^i is the measured value of air parameter 1 at the local station i ; ms_2^i is the measured value of air parameter 2 at the local station i ; ms_m^i is the measured value of air parameter m at the local station i .

The abilities, opportunities and commitment strategies of the agents are taking into consideration during task allocation step. Hence the AC is responsible for the general report design and the ALS_i for each local report i . Using the actions described in table 2 an action plan is given by equation (7)

$$P = \langle\langle \text{create} (\text{local_report}_1\text{_from} (ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1)), ALS_1 \rangle, \langle \text{create} (\text{local_report}_2\text{_from} (ms_1^2, ms_2^2, ms_3^2, \dots, ms_m^2)), ALS_2 \rangle, \dots, \langle \text{create} (\text{local_report}_n\text{_from} (ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n)), ALS_n \rangle, \langle \text{create} (\text{general_report_from} (SM_1, SM_2, \dots, SM_n)), AC \rangle \rangle \quad (7)$$

A social commitment of this group is given by the equation (8).

$$COMM (ALS_1, AC, \text{create} (\text{local_report}_1\text{_from} (ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1))) \wedge COMM (ALS_2, AC, \text{create} (\text{local_report}_2\text{_from} (ms_1^2, ms_2^2, ms_3^2, \dots, ms_m^2))) \wedge \dots \wedge COMM (ALS_n, AC, \text{create} (\text{local_report}_n\text{_from} (ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n))) \quad (8)$$

$$(ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1)) \wedge \dots \wedge COMM (ALS_n, AC, \text{create} (\text{local_report}_n\text{_from} (ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n))) \wedge COMM (AC, \bigwedge_{i=\overline{1, n}} ALS_i, \text{create} (\text{general_report_from} (SM_1, SM_2, \dots, SM_n))) \quad (8)$$

where

$COMM (ALS_1, AC, \text{create} (\text{local_report}_1\text{_from} (ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1)))$ - agent ALS_1 is committed to agent AC to create local report 1

$COMM (ALS_2, AC, \text{create} (\text{local_report}_2\text{_from} (ms_1^2, ms_2^2, ms_3^2, \dots, ms_m^2)))$ - agent ALS_2 is committed to agent AC to create local report 2

$COMM (ALS_n, AC, \text{create} (\text{local_report}_n\text{_from} (ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n)))$ - agent ALS_n is committed to agent AC to create local report n

$COMM (AC, \bigwedge_{i=\overline{1, n}} ALS_i, \text{create} (\text{general_report_from} (SM_1, SM_2, \dots, SM_n)))$ - agent AC is committed to all local ALS_i to create the general report.

In order to create the general report referring to the air quality index there must be designed the local reports. For all n monitoring stations groups of agents the action plan is similar to <MSPD> plan.

3.5 The <MSPD> plan

A monitoring station plan design (MSPD) refers to the action that has to be performed locally at the station level in order to be created the local air quality report. Each ALS_i sets his goal as to design the local report, as in equation (9). Similar to AC, the ALS_i agent cannot make the report on its own, equation (10), and searches among the agency some other agents that can help him to accomplish his goal.

$$GOAL(ALS_i, \text{create} (\text{local_report}_i\text{_from} (ms_1^i, ms_2^i, ms_3^i, \dots, ms_m^i))) \quad (9)$$

$$\neg \text{can}(ALS_i, \text{create} (\text{local_report}_i\text{_from} (ms_1^i, ms_2^i, ms_3^i, \dots, ms_m^i))) \quad (10)$$

So, for each local station i a group G_i of cooperative agents is formed, equation (11), and a common intention is settled, equation (12).

$$G_i = \{ ALS_i, AL_{i1}, AL_{i2}, \dots, AL_{im} \}, \quad i = \overline{1, n} \quad (11)$$

$$C\text{-INT}_{G_i} (\text{create} (\text{local_report}_i\text{_from} (ms_1^i, ms_2^i, ms_3^i, \dots, ms_m^i))) \quad (12)$$

Three main steps may be highlighted for the <MSPD> plan and presented in equation (13): measure parameter j value $j=1,m$, send the measured data at the local station to the ALS_i , create local report i , store the recorded values in databases.

$P_i = \langle \langle \text{measure}(\text{parameter}_1(\text{SM}_i), \text{AL}_{i1}), \text{AL}_{i1} \rangle, \langle \text{measure}(\text{parameter}_2(\text{SM}_i), \text{AL}_{i2}), \text{AL}_{i2} \rangle, \dots, \langle \text{measure}(\text{parameter}_m(\text{SM}_i), \text{AL}_{im}), \text{AL}_{im} \rangle, \langle \text{send}(\text{parameter}_1(\text{SM}_i, \text{ALS}_i), \text{AL}_{i1}), \text{AL}_{i1} \rangle, \langle \text{send}(\text{parameter}_2(\text{SM}_i, \text{ALS}_i), \text{AL}_{i2}), \text{AL}_{i2} \rangle, \dots, \langle \text{send}(\text{parameter}_m(\text{SM}_i, \text{ALS}_i), \text{AL}_{im}), \text{AL}_{im} \rangle, \langle \text{store}(\text{database}_i(\text{date}, \text{hour}, \text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i)), \text{ALS}_i \rangle, \langle \text{create}(\text{local_report}_i_from(\text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i)), \text{ALS}_i \rangle \rangle \rangle$ (13)

The social commitment of group G_i is presented in equation (14).

$\text{COMM}(\text{AL}_{i1}, \text{ALS}_i, \text{measure}(\text{parameter}_1(\text{SM}_i))) \wedge \text{COMM}(\text{AL}_{i1}, \text{ALS}_i, \text{send}(\text{parameter}_1(\text{SM}_i, \text{ALS}_i))) \wedge \text{COMM}(\text{AL}_{i2}, \text{ALS}_i, \text{measure}(\text{parameter}_2(\text{SM}_i))) \wedge \text{COMM}(\text{AL}_{i2}, \text{ALS}_i, \text{send}(\text{parameter}_2(\text{SM}_i, \text{ALS}_i))) \wedge \dots \wedge \text{COMM}(\text{AL}_{im}, \text{ALS}_i, \text{measure}(\text{parameter}_m(\text{SM}_i))) \wedge \text{COMM}(\text{AL}_{im}, \text{ALS}_i, \text{send}(\text{parameter}_m(\text{SM}_i, \text{ALS}_i))) \wedge \text{COMM}(\text{ALS}_i, \bigwedge_{j \in G_i} \text{AL}_j, \text{store}(\text{database}_i(\text{date}, \text{hour}, \text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i))) \wedge \text{COMM}(\text{ALS}_i, \text{AC}, \text{create}(\text{local_report}_i_from(\text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i)))$ (14)

where

$\text{COMM}(\text{AL}_{i1}, \text{ALS}_i, \text{measure}(\text{parameter}_1(\text{SM}_i)))$ – agent AL_{i1} is committed to ALS_i to measure the value of parameter 1 from monitoring station i ;

$\text{COMM}(\text{AL}_{i1}, \text{ALS}_i, \text{send}(\text{parameter}_1(\text{SM}_i, \text{ALS}_i)))$ - agent AL_{i1} is committed to ALS_i to send the recorded value for parameter 1 at station i to ALS_i ;

$\text{COMM}(\text{AL}_{i2}, \text{ALS}_i, \text{measure}(\text{parameter}_2(\text{SM}_i)))$ – agent AL_{i2} is committed to ALS_i to measure the value of parameter 2 from monitoring station i ;

$\text{COMM}(\text{AL}_{i2}, \text{ALS}_i, \text{send}(\text{parameter}_2(\text{SM}_i, \text{ALS}_i)))$ - agent AL_{i2} is committed to ALS_i to

send the recorded value for parameter 2 at station i to ALS_i ;

$\text{COMM}(\text{AL}_{im}, \text{ALS}_i, \text{measure}(\text{parameter}_m(\text{SM}_i)))$ – agent AL_{im} is committed to ALS_i to measure the value of parameter m from monitoring station i ;

$\text{COMM}(\text{AL}_{im}, \text{ALS}_i, \text{send}(\text{parameter}_m(\text{SM}_i, \text{ALS}_i)))$ - agent AL_{im} is committed to ALS_i to send the recorded value for parameter m at station i to ALS_i ;

$\text{COMM}(\text{ALS}_i, \bigwedge_{j \in G_i} \text{AL}_j, \text{store}(\text{database}_i(\text{date}, \text{hour}, \text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i)))$ – agent ALS_i is committed to all local agent from group G_i to store the received recorded data into specialized local database;

$\text{COMM}(\text{ALS}_i, \text{AC}, \text{create}(\text{local_report}_i_from(\text{ms}_1^i, \text{ms}_2^i, \text{ms}_3^i, \dots, \text{ms}_m^i)))$ -agent ALS_i is committed to AC to create the local report i .

3.6 The AQDMAS general algorithm

Based on the two types of action plans described in previous sections the general running algorithm of the AQDMAS system follows the next steps:

1. The coordinator agent asks the needed resources for the general air quality report design to the monitoring stations supervisors;
2. The local supervisor agents initiate requests to the local agents to provide the measured data for needed atmospheric parameters;
3. The local agents measure the concentrations values for local air parameters and send these values to the local supervisor agents;
4. After creating the local report (LR) the ALS_i send it to the AC in order to be design the general air quality report (GR).

4. Experimental Results

Zeus Agent Building Toolkit, a Java-based toolkit for intelligent agents' development was used in order to implement a preliminary agent based prototype system. Zeus allows portability, which is assured by the use of Java, and defines an agent platform that may be distributed across a network. *Visualiser* agent is optional but very useful because it provides a possible interface

between the user and the system. In this case study, there were designed seven intelligent agents, placed on different computers.

Table 3. The AQDMAS general algorithm

```

Begin
While  $\neg$  create(GR) do
  initiate_request(resources_for(AC, GR)
  initiate_request (resources_for (ALSi, LRi)
begin parallel execution
  for j=1 to m do
    if  $\neg$  is_working( $T_j^1$ ) then
       $ms_j^1$  =get(measure( parameterj(SM1)))
    else  $ms_j^1$  =null
    send(parameterj(SM1, ALS1)
  end for
store(database1(
  data, ora,  $ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1$ ))
create(local_report1_from(
   $ms_1^1, ms_2^1, ms_3^1, \dots, ms_m^1$ ))
||
.....
||
for j=1 to m do
  if  $\neg$  is_working( $T_j^n$ ) then
     $ms_j^n$  =get(measure(parameterj (SMn)))
  else  $ms_j^n$  =null
  send(parameterj (SMn, ALSn)
end for
store(databasen (
  data, ora,  $ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n$ ))
create(local_reportn_from (
   $ms_1^n, ms_2^n, ms_3^n, \dots, ms_m^n$ ))
end parallel execution
create(general_report_from(SM1, SM2, ..., SMn))
end while
end

```

For each project developed, there are generated task agents (created by the project developer with tasks specific to the application) and at least three utility agents: Name Server, Facilitator and Visualiser. The *Name Server* and *Facilitator* agents are essential for the system because they are the ones who know the IP addresses and the tasks that can be performed by each agent individually. The *Visualiser* agent is essential for the system because they are the ones who know the IP addresses and the tasks that can be performed by each agent individually. These machines are located in the Department of Information Technology, Mathematics and Physics, at the Petroleum-Gas University of Ploiești in Romania. Although the machines have different hard drives, the same set of files created with Zeus is shared on every one of them. This is to ensure that all the related initialization and configuration files are accessible from every machine. The TCP/IP protocol is used to transfer data in our system. It uses packet switching to transmit data between the hosts A, B, and C. When the packets reach their destination, they are reassembled in the proper order. The version of IP used is Internet Protocol Version 4 (IPv4), which uses 32-bit addresses [11]. Table 4 presents the agent distribution among these three hosts.

In this experiment the system is tested as a simulation for n=2 monitoring stations and m=10 local parameters analyzed: the measurements are already done and the agents extract from databases the needed values.

4.1 Data set

The data set used represents the recorded data for major air pollutants at two monitoring stations (PH1, and PH3) located in Ploiesti, Romania. The two stations are part of the National Air Quality Monitoring Network of Romania which consists of 117 stations placed in 37 counties for monitoring the influence of the traffic, industrial activity and urban settlements over the air quality [19]. The measurements are made hourly, at each station, and after a primary validation they are stored in dedicated databases. They are also published online on www.calitataer.ro for the population information regarding a possible pollution episode.

The records of carbon monoxide (CO), sulphur dioxide (SO₂), nitrogen monoxide (NO), nitrogen dioxide (NO₂), particular matter (PM₁₀) and ozone (O₃) concentrations from January 2011 until December 2011 are used for this simulation.

Table 4. Agent distribution among hosts

	Host A	Host B	Host C
Coordinator agent	x		
Local supervisor 1		x	
Local supervisor 2			x
Local agent 11		x	
Local agent 12		x	
Local agent 21			x
Local agent 22			x
Name Server	x		
Visualiser	x		
Facilitator	x		

4.2 Simulation

At the running time the AQDMAS system GUI interface contains the information about the monitored air quality parameters (see in Figure 1). After being selected the day required for the simulation, the Coordinator Agent starts to collect all the necessary information to complete a general air quality report. Thus, he requests the local reports to the Local Supervisor agents. In order to fulfil the requirement they ask the local agents to extract from the specialized MySQL databases the data needed. One of the local agents has access to the pollutant parameters data recorded such as: NO, NO₂, SO₂, CO, PM₁₀, and O₃, and the other one to the meteorological parameters that have great influence over a possible pollution episodes such as: precipitation, temperature, wind speed, and wind direction.

Once extracted all the required data, the Local supervisor agents compare the values with the lower and upper limits of the intervals presented in system interface and establish which is the local air quality index (as a maximum among the parameters indexes). The local reports are therefore sent to the coordinator agent, which will create the general report with the general air quality index as maximum between the local indexes. Figure 1 presents the system GUI interface. In this experiment at the monitoring station 1 the NO₂ concentration is 112.83mg/m³ which is high and according to the first table in interface makes that the local air quality index to be 3 (good). The general air quality index is 3 and represents a medium level of pollution.

Several reports and statistics are provided by the Visualiser agent in order to analyze the agents' behaviour and the functioning of the AQDMAS system. One facility offered by the Visualiser agent is the possibility of tracking the task status. In Figure 2 is presented an intermediate state of the process in the Report Tool. Thus, the completed tasks are colored with white (Extract *Meteo Data1*, Extract *Pollutant Data1*, Extract *Meteo Data2*, Extract *Pollutant Data2*) and the green ones are the tasks that are currently calculating the local air quality index (Create *Local Report1* and Create *Local Report2*). The Create *General Report* task is coloured in blue meaning that it is firm.

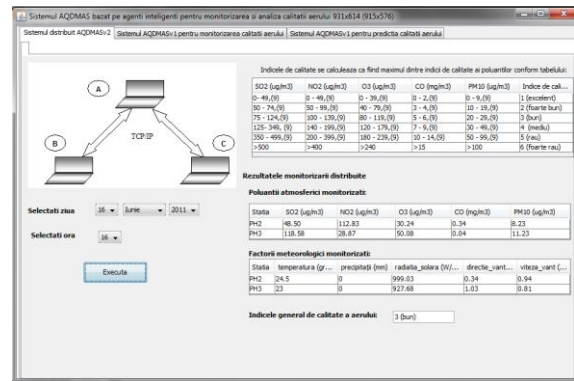


Figure 1. Screenshot of the AQDMAS system GUI interface

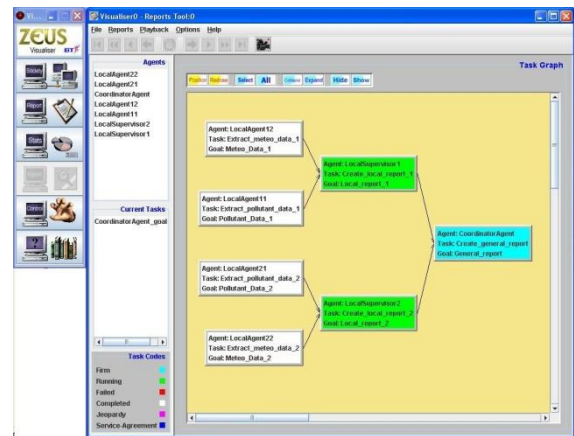


Figure 2. Screenshot of the task graph obtained with the Visualiser agent during the AQDMAS system run

There can be created a series of statistics related to the collaboration and the distribution by type or by agent of the traffic volume. Two detailed statistics that highlights the collaboration among the agents are presented in figure 3, because 63% from all the messages that were sent are to inform (in order to achieve the general goal, the agents send the information requested). Furthermore, the subscribe messages are 4% (to ask the Facilitator

which task is performed by which agent), call for propose 9% and propose 9% (the supervisor agent ask the other agents to send the information needed), and 9% of the messages are accept-proposal type (with no negotiation the agents accept to fulfil the request).

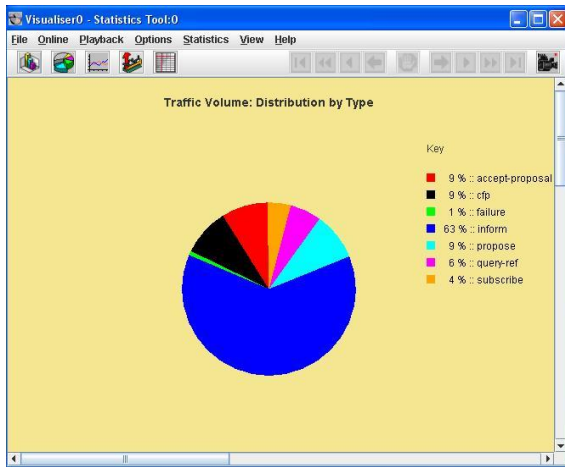


Figure 3. The traffic volume- distribution by type of AQDMAS System in Zeus

5. Conclusions

This paper presented the BDI logic based model design and the implementation of a multi-agent based distributed system, AQDMAS, for monitoring and analyzing air quality. The experimental system consists in a set of three types of agents distributed on three computers that simulates two monitoring stations and another one for the coordination centre. The multi agent system is implemented in FIPA compliant ZEUS open source platform. The results of the simulations show the effectiveness of the proposed monitoring and analyzing technique and the possibility of autonomous built-in simulation of tracking the air quality evolution with distributed resources. The environmental monitoring success is better accomplishable through distributed monitoring of the environment state. Distributed multi-agents sustain such need through sensors and actuators, and subsequent collaboration. Other advantages of agents based system include support for a more flexible peer-to-peer (P2P) model, reduction of network bandwidth use and scalability.

Future works includes building models where more intelligence is added to the agents, so that they may autonomously organize themselves to better manage the monitoring process and to improve the methodology for the pollution episode prediction. Another direction involves

the integration of a measurement subsystem formed by sensors and transducers for the real time data acquisition in the system.

REFERENCES

1. CHAPPIN, E. J. L., G. P. J. DIJKEMA, **An Agent based Model of the System of Electricity Production Systems: Exploring the Impact of CO2 Emission-trading**, Proceedings of IEEE International Conference on System of Systems Engineering, IEEE, San Antonio, TX, USA, 2007, pp. 1-5.
2. COLLINS, J., D. NDUMU, **The Application Realisation Guide**, The Zeus Agent Building Toolkit, Intelligent Systems Research Group, BT Labs, 1999.
3. DENZINGER, J., H. KASINGER, B. BAUER, **Decentralized Real-time Control of Water Distribution Networks using Self-organizing Multi-agent Systems**, SASO, 2010, 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, pp. 223-232.
4. DRAGOMIR, E. G., M. OPREA, **A Multi-Agent System for Power Plants Air Pollution Monitoring**, Proceedings of 2nd IFAC Workshop Convergence of Information Technologies and Control Methods with Power Systems, 2013, pp. 97-102.
5. FAGIN, R., J. HALPERN, Y. MOSES, M. VARDI, **Reasoning about Knowledge**, MIT Press, Cambridge, 1995, MA, USA.
6. HUANG, W., X. ZHANG, J. WANG, **Multi-agent System Computing and Simulation of Inter-basin Water Transfer**, Intelligent Automation & Soft Computing, vol. 17(7), 2011, pp. 897-908.
7. KEPLICZ, B. D., R. VERBRUGGE, **Teamwork in Multiagent Systems- A Formal Approach**, John Wiley & Sons LTD, 2010.
8. KHOO, H. K., Q. MENG, **Modeling of Vehicle Emission Pricing Strategy Using Multi-agent System**, Journal of the Eastern Asia Society for Transportation Studies, Vol. 8, 2010, pp. 883-895.
9. MEYER, J.-J. C., W. VAN DER HOEK, **Epistemic Logic for AI and Theoretical**

- Computer Science**, Cambridge University Press, 1995, Cambridge, UK.
10. MENG, A., L. YE, D. ROY, P. PADILLA, **Genetic Algorithm based Multi-agent System Applied to Test Generation**, *Computers and Education*, vol. 49(4), 2007, pp. 1205-1223, ISSN 0360-1315.
 11. MORLEY, D., C. S. PARKER, J. LAVINE, **Understanding Computers: Today and Tomorrow**, 10th edition, Thomson Course Technology, 2004.
 12. NI, J., L. REN, M. LIU, D. ZHU, **A Multiagent Dynamic Assessment Approach for Water Quality based on Improved q-Learning Algorithm**, *Mathematical Problems in Engineering*, vol. 2013, Article ID 812032, 2013, 7 pages, doi:10.1155/2013/812032.
 13. OPREA, M., C. NICHITA, **On the Distributed Water Pollution Control Solving with an Agent-Based Approach**, *Studies in Computational Intelligence*, Springer, 2008, pp. 289-294.
 14. OPREA, M., A. MATEI, E. PETRE, **Agent-based Modeling of a Dam Monitoring System**, *Proceedings of CSCS17*, vol. 2, 2009, pp. 509-514.
 15. OPREA, M., M. CĂRBUREANU, E. G. DRAGOMIR, **AirQMAS: A Collaborative Multi-agent System for Air Quality Analysis**, *Annals of the University of Craiova, series Automation, Computers, Electronics and Mechatronics*, vol. 9(37), No 1, 2012.
 16. PAPALEONIDAS, A., L. ILIADIS, **Hybrid and Reinforcement Multi agent Technology for Real time Air Pollution Monitoring**, 8th IFIP WG 12.5 International Conference, AIAI 2012, Halkidiki, Greece, 2012, pp. 274-284.
 17. PETRE, E., **AgentAirPol System, an Agent-based System for Air Pollution Analysis**, *Bulletin of PG University of Ploiești, Series Mathematics, Informatics, Physics*, vol. LX, nr. 1, 2008, pp. 63-72.
 18. RALHA, C. G., C. V. S. SILVA, **A Multi-agent Data Mining System for Cartel Detection in Brazilian Government Procurement**, *Expert Systems with Applications*, vol. 39, 2012, pp. 11642-11656.
 19. Reteaua Nationala de Monitorizare a Calitatii Aerului, www.calitateaer.ro
 20. ROCHE, R., L. IDOUMGHAR, S. SURYANARAYANAN, M. DAGGAG, C. A. SOLACOLU, A. MIRAOUI, **A Flexible and Efficient Multi-agent Gas Turbine Power Plant Energy Management System with Economic and Environmental Constraints**, *Applied Energy*, vol. 101, 2013, pp. 644-654
 21. VAN OEL, P. R., M. S. KROL, A. Y. ARJEN, Y. HOEKSTRA, R. R. TADDEI, **Feedback Mechanisms between Water Availability and Water Use in a Semi-arid River Basin: A Spatially Explicit Multi-agent Simulation Approach**, *Environmental Modelling & Software*, vol. 25, no. 4, 2010, pp. 433-443.
 22. WEISS, G., **Multi-agent Systems – A Modern Approach to Distributed Artificial Intelligence**, The MIT Press, Cambridge, 1999.
 23. WOOLDRIDGE, M., **An Introduction to Multi-agent Systems (2nd ed.)**. Wiley Publishing, 2009, ISBN 0470519460, 9780470519462.
 24. WU, S., P. NOY, **A Conceptual Design of a Wireless Sensor Actuator System for Optimizing Energy and Well-being in Buildings**, *Intelligent Buildings International*, 2(1), 2010, pp. 41-56.