**MS-Windows Monitoring Based Cooperative Agents**

**Abbass M. Al Bakry**

*Babylon UniversityCollege of Information Technology*

**Marwa Khidir Obayes**

*Babylon University Faculty of sciencesComputer science Department*

**Abstract**

Because of the need to increase reliability and decrease failure in machines, we developed software agent that monitors important files that dedicated with the different applications and affect the functioning and integrity of the windows. The designed software agent is for monitoring the authorization and authentication of the windows system users, to observe data in applications files and to avoid failure in them. This task done by keeping track the effects that occurred on these files and the necessary events can be saved on the online development DB. The development DB can be shown to the administrator as development report with all events happened for the dedicated files to be taken in account.

**الخلاصة**

بسبب الحاجة لزيادة الوثوقية وتقليل الفشل في الأجهزة، طورنا نظام وكيل يراقب الملفات الهامة المخصصة لتطبيقات مختلفة وتؤثر على سير وسلامة نظام الوندوز. نظام الوكيل المصمم هو لمراقبة التخويل والوثوقية لمستخدمي نظام الويندوز، لمراقبة البيانات في ملفات التطبيقات وتجنب الفشل فيها. هذه المهمة تحصل من خلال تعقب التأثيرات التي وقعت في هذه الملفات ويمكن حفظ الأحداث الضرورية في قاعدة بيانات تطور آنياً. يمكن عرض قاعدة البيانات المطورة لمدير النظام على هيئة تقرير يحوي كل الأحداث التي حدثت للملفات لأجراء اللازم.

**Introduction**

Cooperative software agents in multi agent systems have recently attracted a lot of attention and have been experimented in building a variety of applications.

The term of an *agent* has recently become important in Artificial Intelligence (AI), and newly subfield, Distributed AI (DAI).

Intelligent agents research began around the mid of 1980s, but did not gain the widespread attention of technologists until it was spotlighted by a popular computing journal, *Communications of the Association of Computing Machinery*, in 1994.

Intelligent agents represent the convergence of technologies and methods such as object orientation, components, distributed computing, DAI and human-computer interface design.

**Basic Concepts:**

**Monitoring:** is the process of gathering, collecting, and presenting information about observing systems or objects. [Yigal94]

**Agent:** anything can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. [Russel95]

**Cooperative agents:** autonomous agents which cooperate for mutual economic and management benefit.

**Historical Background:**

Agent has been used in several domains using a variety of technical fields, as following:

* ***(D.Stipaničev, M.Štula, D.Krstinić, and Lj.Šerić , 2003) iForestFire:***

It was practical implementation of the observer network system, developed for surveillance forests to detect fires in the forests.

The effective way to minimize damage caused by forest fires is their early detection and fast reaction, apart from preventive measures.

* ***(Matthew L. Massie, Brent N. Chun, and David E. Culler, 2004) Ganglia:***

It is scalable distributed monitoring system for high performance computing systems such as clusters and grids.

* ***(Microsoft Windows ) Pc Pandora:***

It is great monitoring software to track internet and computer activity on any computer in your home. It offers a host of effective monitoring tools, blocking applications and reporting capabilities.

* ***(Christina B. Vilakazi and Tshilidzi Marwala, 2007):***

This paper demonstrated a design of a condition monitoring system using intelligent agents and multi-agent systems.

A functional design of the condition monitoring consists of four layers: Data monitoring layer, interpretation layer, diagnostic layer, and information layer.

Data monitoring consists of two agents for extracting features and conditioning of data received from the measurement system.

Interpretation layer consists of three agents: kernel-based agent, back propagation agent, and extension neural network agent,

These agents interpret data from the data monitoring agent and give a diagnosis on the condition of the equipment.

The diagnosis layer uses weighted majority voting to combine the decision from the interpretation agent

* ***(Ammal Hanash, 2002):***

This thesis designs a monitoring system based on an agent technology and applied in a Local Area Network (LAN) environment

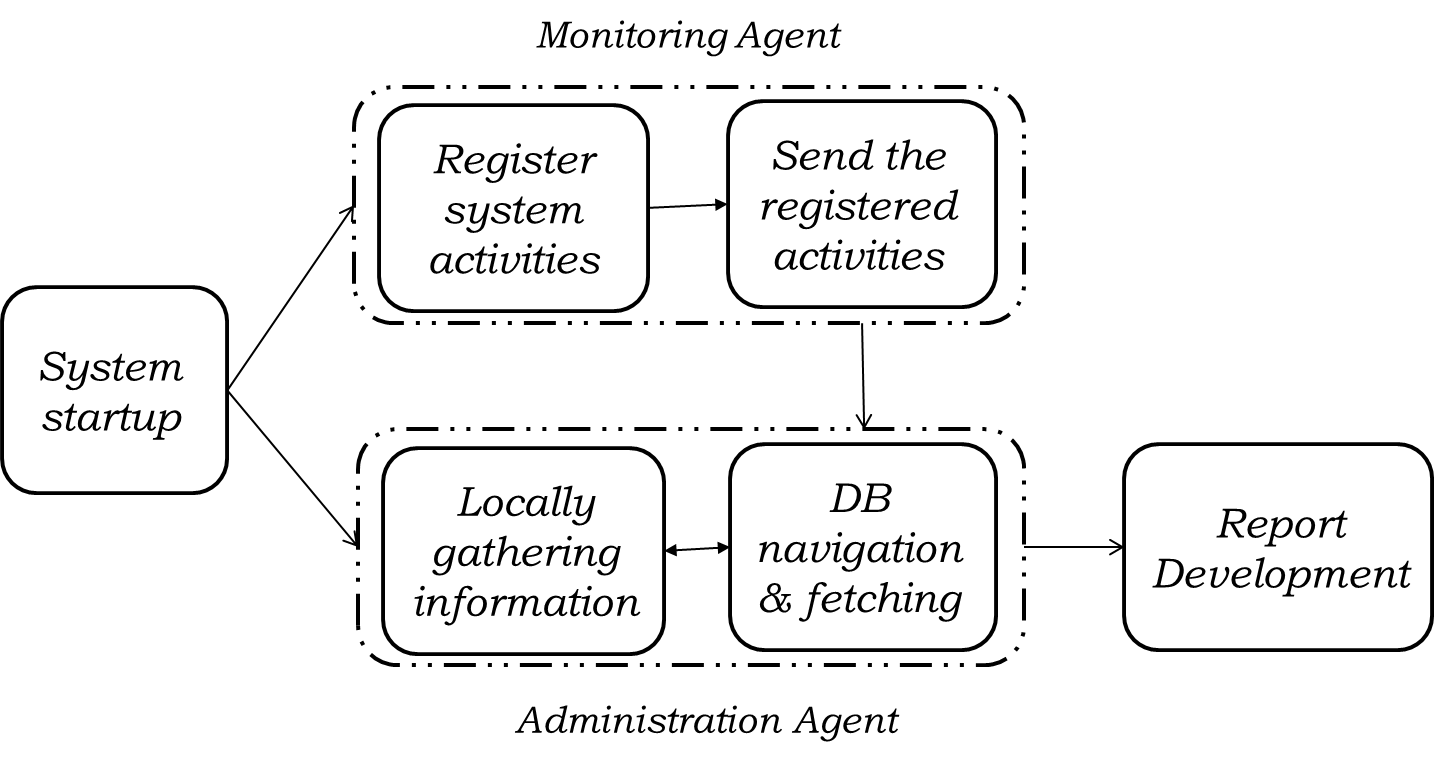
**MAS System Description:**

Monitoring agent system (MAS) consists of two types of agents: *Administration agent (admin)* and *Monitoring agent (user)*.

Admin agent resident in the centralized computer and control on other Monitoring agents, and Monitoring agent resident in client's computers, both of them will work at startup of windows system.

Admin agent must work firstly in MAS System, so it enables to receive starting signals from monitoring agents.

We will interest with some features of the specific files, like *full path, size, last modified*.



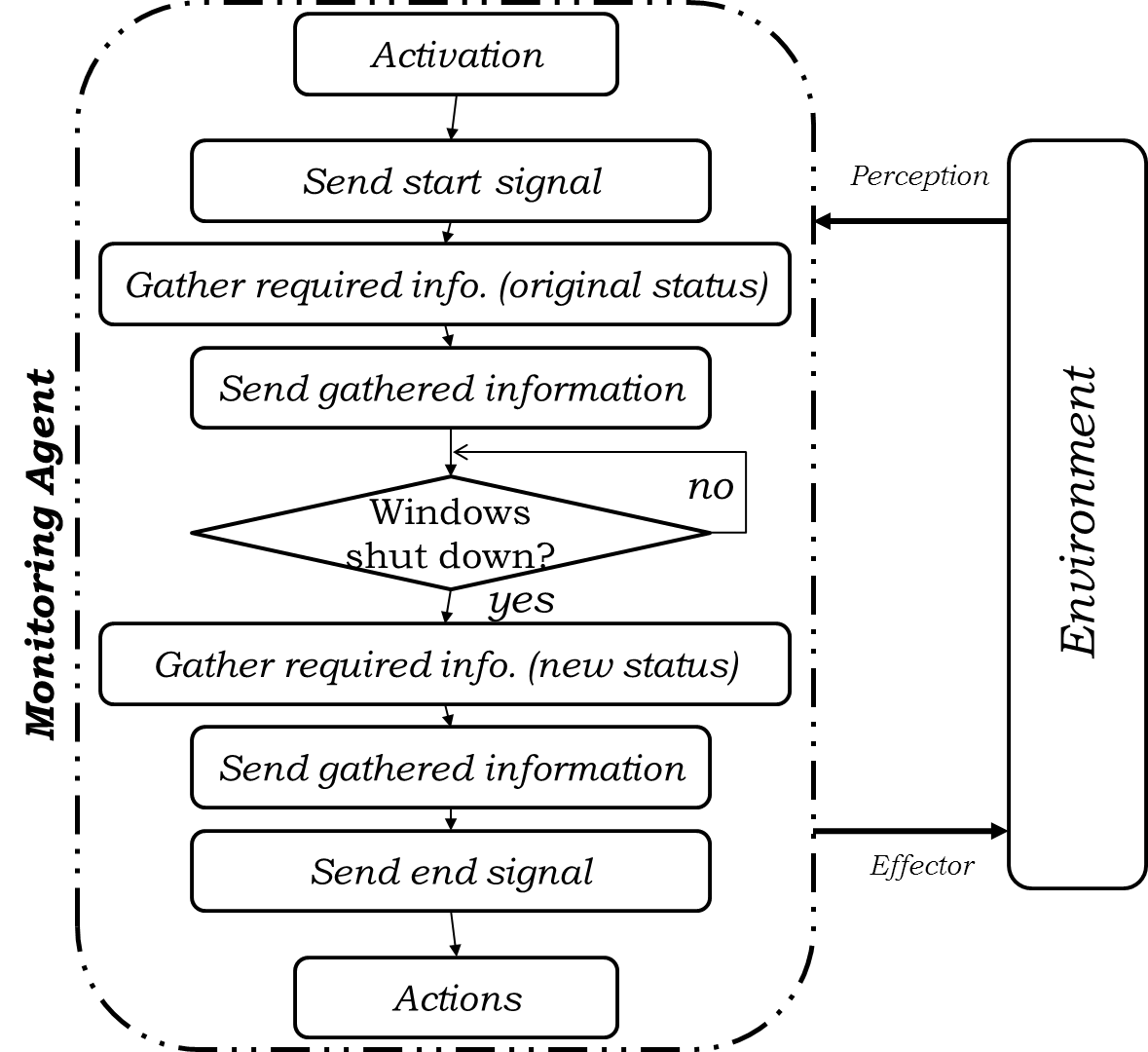
*Figure (1): MAS System Architecture*

**Monitoring agent Architecture:**

When monitoring agent starts silent working, it send starting signal to the admin agent to notify it that this agent is online at this moment. Then, it takes the specific information from its environment -as perception- and sending them to the admin agent (as original status of files).

Then, when windows system shut down, the agent acquired new perception (as new status of files) and send them to admin agent, too.

Finally, send ending signal to admin agent to notify it by shutting down this user computer. Figure (2) shows Monitoring agent architecture.



*Figure (2): Monitoring agent Architecture*

**Administration Agent Architecture:**

When admin agent starts working, start off listening to monitoring agents, in addition monitor local environment.

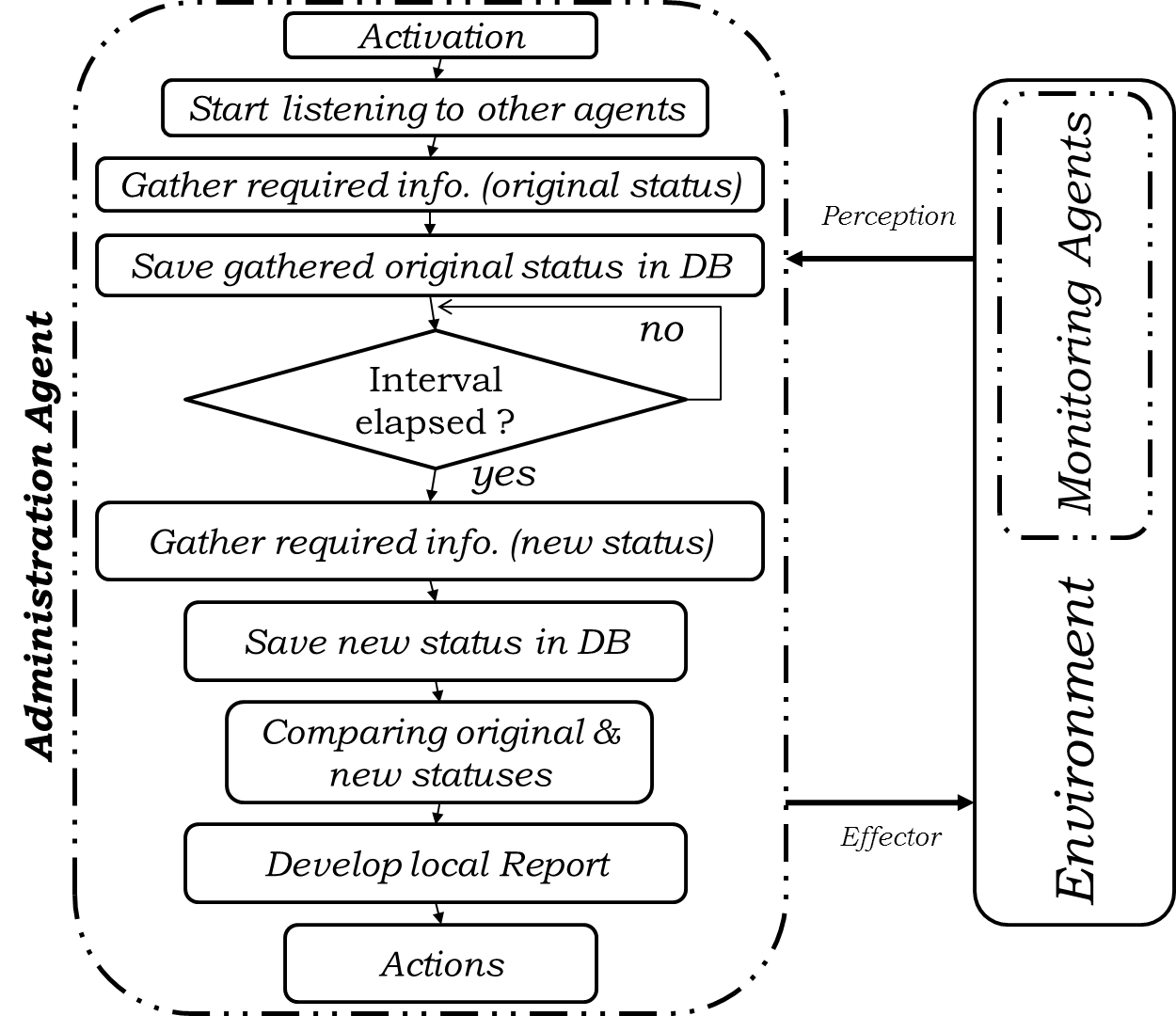
So, after receiving starting signal from any monitoring agent, it awaits receiving the specific information.

When it receives them, it saves them in DB (as original status of files), and in shutting down this user computer, it receives new information (as new status of files) followed by ending signal.

At this moment, admin agent compares between original and new statuses to develop final report of this session.

Locally monitoring summarized as follows:

At starting windows system, agent gathers the specific information (as original status of files), and -after limited period- gathers new information (as new status of files) and compares them to develop final local report. Figure (3) shows Admin Agent architecture.

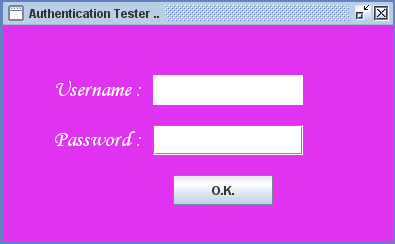


*Figure (3): Admin Agent Architecture*

**Case Study:**

The implementation of MAS System starts at startup of windows system working and finishes its running at shutting down it. It is supposed that there is one Administration agent (run firstly) and many of Monitoring agents to observe them, both of them runs at startup of windows.

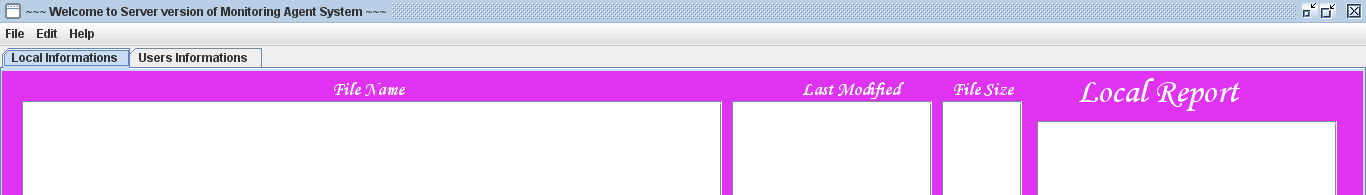
The working of our system is silent, therefore at first sight, the GUI do not show (working in background), and only the administrator has the accessibility to the GUI of administration agent and Monitoring agents by inserting special username and password to show it. Figure (1) shows the *Authentication Tester*.



*Figure (4): Snapshot of Authentication Tester*

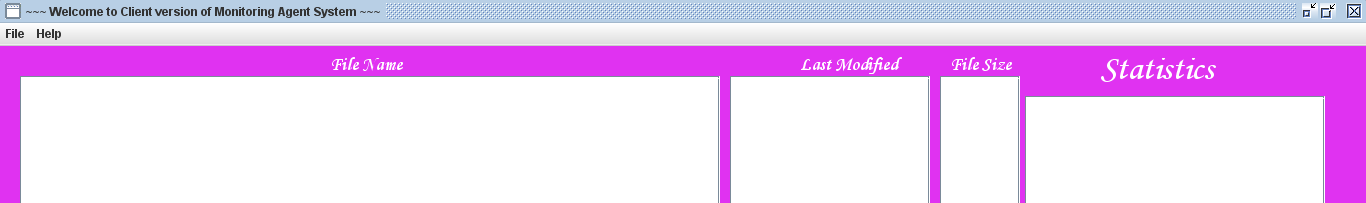
After inserting correct username and password the general GUI will appear, as follows:

Figure (4) shows Administration Agent GUI.



*Figure (5): Snapshot of Administration Agent GUI*

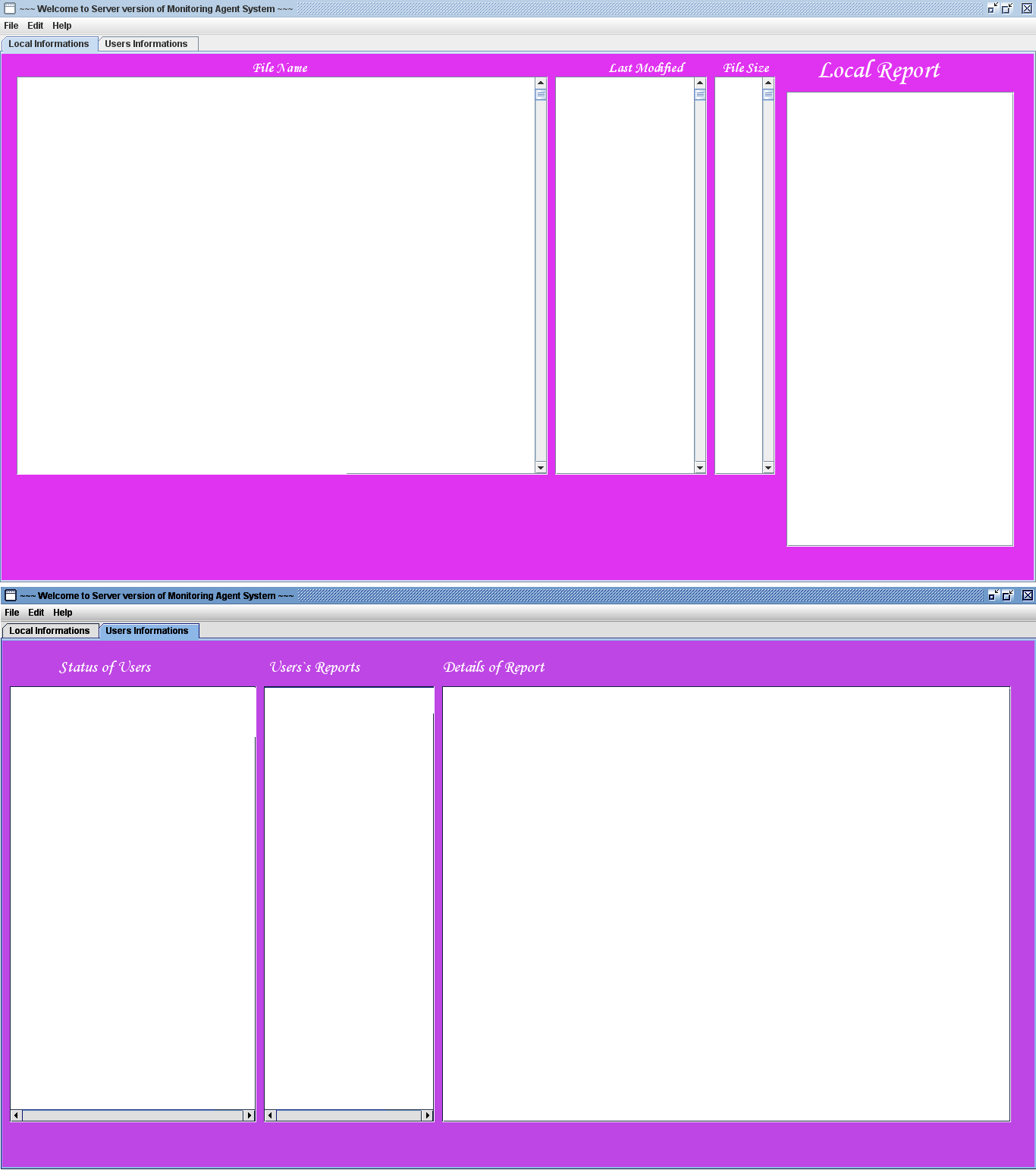
And Figure (6) shows Monitoring Agent GUI.



*Figure (6): Snapshot of Monitoring Agent GUI*

Implementation of Monitoring Agent:

As mentioned above, the Administration Agent does two things Local Monitoring and Remote Monitoring. Because of the administration agent in addition to monitoring local events, it is listening to other agents and receiving their information for storing and comparing them, its general window consists of two parts: *Local information* (contains information resulted from local monitoring) and *User`s information* (contains information receiving from another monitoring agents). Figure (7) shows the two parts of Administration Agent.

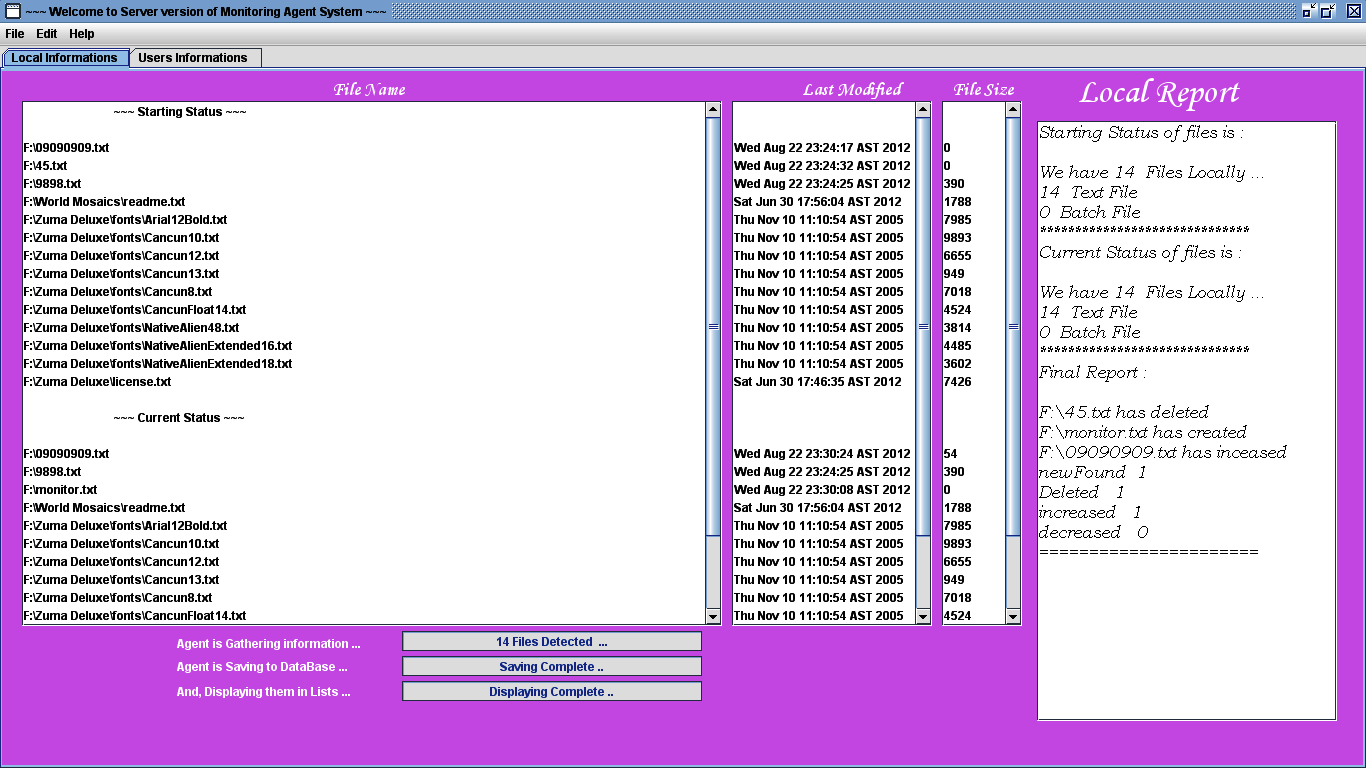


*Figure (7): Snapshots of Administration Agent Parts*

1. *Local Monitoring:*

In Local Monitoring, Administration Agent starts gathering the specific information from its local environment by traversing the file system (up-down approach) searching on any files that have extension (.txt or .bat) which if any faults happened with them will affect the windows file system, and stores them in MySQL Data Base, then retrieves them from DB for showing them in lists, preceded by the words "Starting Status".

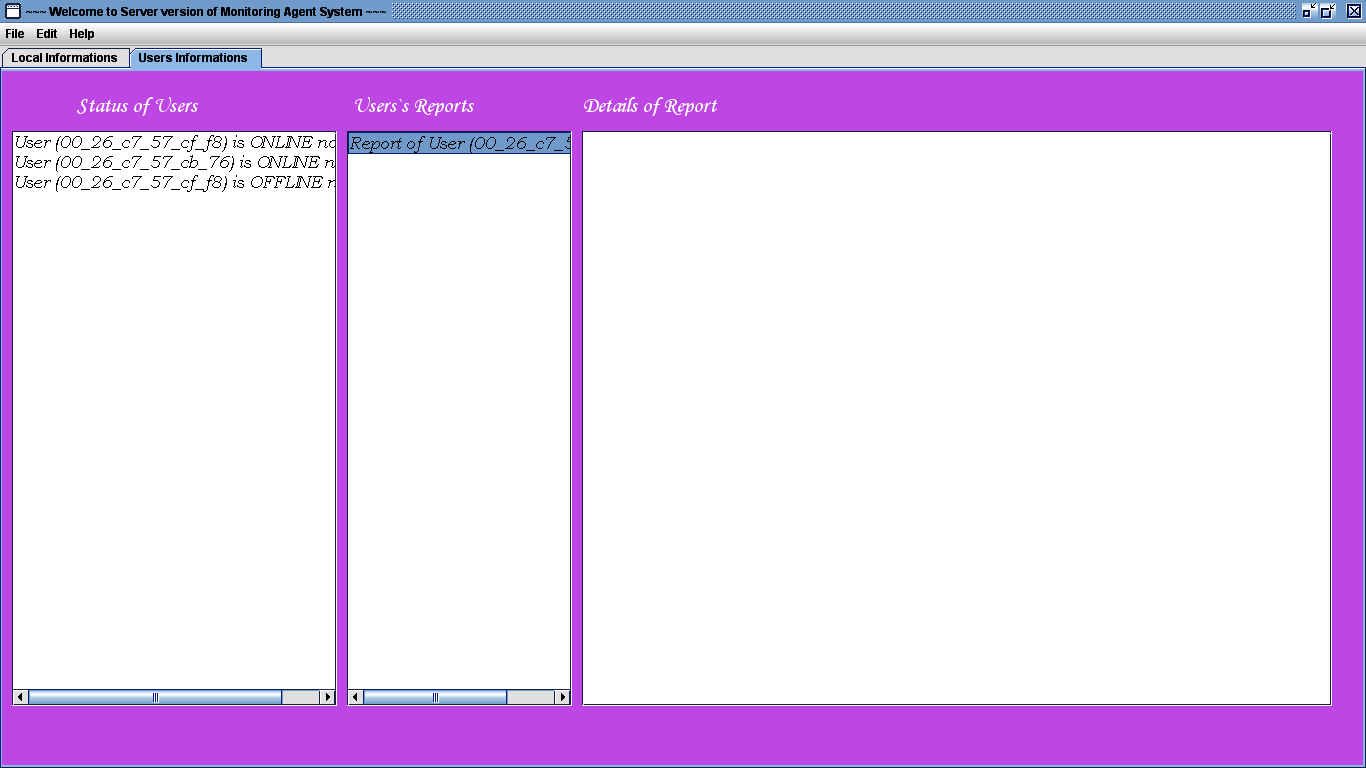
And, after a specified period, will repeat its previous processes of gathering, storage and display again in the lists, preceded by words "Current Statuses" then begin comparing process, which compares between these two created tables in the Data Base to prepare the full report of this session. As shown in Figure (8).



*Figure (8): Snapshot of Administration Agent GUI (Local Monitoring Window)*

1. *Remote Monitoring:*

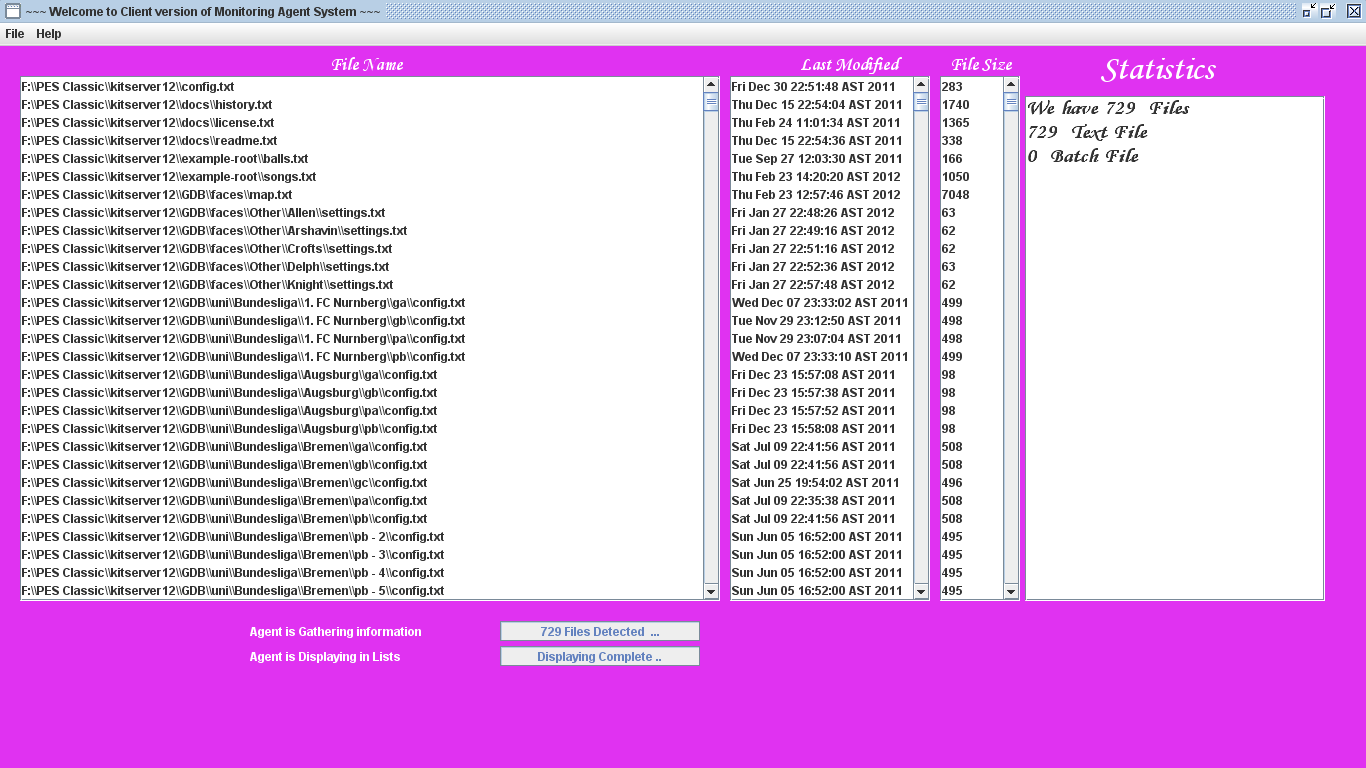
While in Remote Monitoring, the *monitoring agents* gather the specific information from them environments and send them each with the name of its agent to Administration agent to store them in the Data Base of that agent in table named of time of sending these information and then at shutting down of windows in that agent, it gathers new information again and then compare between the two statuses of files before working this computer and these at shut down. As shown in Figure (9).



*Figure (9): Snapshot of Administration Agent GUI (Remote Monitoring window)*

Implementation of Monitoring Agent:

Implementation of Monitoring Agent starts at startup of windows system, as mentioned previously. At this moment, it gathers the specific information from its local environment, and prepares name for this session to send it with the information to Administration Agent to use it as name of this connecting monitoring agent (from the MAC address of its machine) which is used as Data Base name for storing information of this monitoring agent, then sends these information and agent`s name (MAC address) and table name in that Data Base (from current time for starting this session followed by word "\_Start") to the Administration agent step by step. Figure (10) shows Monitoring Agent (Gathering the information of its environment).



*Figure (10): Snapshot of Monitoring Agent (Gathering information)*

And at shutting down of windows system, the Monitoring Agent will repeat the same steps for gathering and sending information (as new statuses), where it gathers the specific information again and sending them with the same Data Base name (MAC address of this machine) and the same table name (time of starting that session) followed by word "\_End" to the Administration Agent for storing them and executing the comparison process.

**Conclusions**

Because of the need to increase machine reliability and decrease machines failure, so monitoring the work of systems takes us away from their failure that could happen.

The designed software agent monitors important files - that dedicated with the different applications and affect the functioning and integrity of the windows- by keeping track the effects that occurred on these files and the necessary events can be saved on the online development DB.

The development DB can be shown to the administrator as development report with all events happened for the dedicated files to be taken in account.

**References**

Yigal Hoffner, **1994**, **Monitoring in Distributed Systems**, Poseidon House - Castle Park.

Ammal Hanash, **2002**, **Agent-based system for monitoring in a Lan Environment**, MSc Thesis, University of Technology, Department of Computer Science.

Russel, S. P. Norvig, **2003**, **Artificial Intelligence: A modern Approach**, Pearson Education.

D.Stipaničev, M.Štula, D.Krstinić and Lj.Šerić, **2003**, **Agent Based Intelligent Forest Fire Monitoring System**, Department for Modelling and Intelligent Systems - Center for Scientific Computing.

[

Matthew L. Massie, Brent N. Chun, David E. Culler, **2004**, **The Ganglia Distributed Monitoring System: Design, Implementation, and Experience**, University of California, Berkeley, Computer Science Division, Berkerley, CA 94720-1776, USA.

Christina B. Vilakazi and Tshilidzi Marwala, **2007**, **Agents and Multi-agent Systems and Application to Condition Monitoring**, paper.

Microsoft Windows, PCPandora - Help System

<http://www.pcpandora.com/>