

# A GENERAL SYSTEM FOR MONITORING AND CONTROLLING VIA INTERNET

Adrián Peñate Sánchez, Ignacio Solinis Camalich  
Alexis Quesada Arencibia and José Carlos Rodríguez Rodríguez  
*Instituto de Ciencias y Tecnologías Cibernéticas, Universidad de Las Palmas de Gran Canaria  
E35017, Las Palmas, Spain*

Keywords: J2EE, Java, Web Monitoring and Controlling, Automation, Mobile Technologies, Internet Computing.

Abstract: The main aim of the project, presented in this paper, is to create a system that will give a solution to the variety of situations we can encounter in industrial plants control; to achieve this goal we have given the system different automation, remote control and remote monitoring functionalities. Industrial installations can be composed of several plants; for this reason we offer a solution capable of defining a central office that is in control of all plants, each plant with its own controller. Apart from the problem of geographically disperse locations, it is also of the essence to offer automation for the tasks that take place in the plants. In order to achieve an automation system capable of giving answer to a range of situations, as wide as possible, we have designed an engine so that the user may create its own rule based system, defining himself the rules by means of an interface that abstracts the user from the formal definition of the rules.

## 1 INTRODUCTION

This application sprung up from the approach of several companies interested in having an application capable of surveying their plants and being able to do those tasks automatically. These approaches gave place to the development of two consecutive projects that gave birth to the system that we will outline in this paper.

### 1.1 Final Objective

The challenge was to create a system by means of which an industrial plant could be controlled remotely and its processes automated; all this without being tied to a specific kind of industrial plant or process.

For this we understand that any industrial process, generally, is composed by three factors:

- Sensors.
- Actuators.
- Decision making.

By intending to control these factors we have settled the basis on which to build the system.

### 1.2 System Structure

The system's structure is composed by two subsystems: the plant and the office; each one of them will be managed by a different application. The different plants will be connected to the office, from where the surveillance of the plants functioning state will be made.

The office will be managed by an application that will take care of the visualization of the plants state, store the historical data of each plant, allow the user to make simple statistical studies of the historical data, etc.

Each plant's application will take care of the usual tasks of monitoring and automation, as well as communicating with the office. The information flow between the office and the plants is bidirectional, being established under the following circumstances.

- The office starts the communication in order to obtain real time data about the plant
- The plant sends warnings about the activation or deactivation of alarms. We indistinctively refer to rules or alarms in the same way.

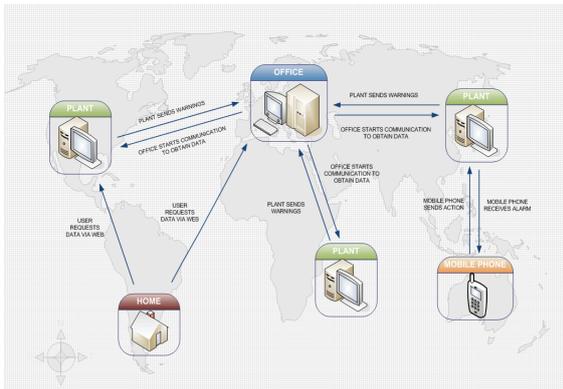


Figure 1: Topology of the system. In this picture both subsystems can be seen in a centralized topology where plants act as peripheral nodes and the office as the central node.

In the image we can appreciate how the system’s different actors start communicating in order to perform the tasks that are required. It can be seen how the office asks the plants for data and how the plants send the historical data to the office. It can also be seen how the mobile phone starts the communication, when it sends an action to the plant, and how the plant starts the communication, when it sends an SMS in the event of an alarm. Another possible communication process can be established in any computer with access to the internet, you can ask both the plants and the office for information about the actual state of the installations.

### 1.3 Development with Open Source Tools

One of our main concerns was to develop this system with Open Source tools. For this reason we defined several requirements in order to use the most suitable tools for the purpose of the project.

After the process of studying and selecting different languages, frameworks and development software, we chose the following ones:

- The developed system is based on a Java EE 5 platform. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitier Web-based applications.
- Open Source software: Linux, Specific software (application server, databases, etc.), Integrated development environment (Eclipse, DBdesigner, etc.).

## 2 SYSTEM ARCHITECTURE

In this section we will introduce the system’s architecture. We are going to show its modular design as well as the most relevant modules.

### 2.1 A Modular Approach to the Problem

This system was conceived as one that was going to be made and evolved in several issues, so, as for the architecture, a modular approach was of essence. Introducing a modular design would make the system capable of adding new modules without other parts needing to be changed or adapted to assemble with the new ones.

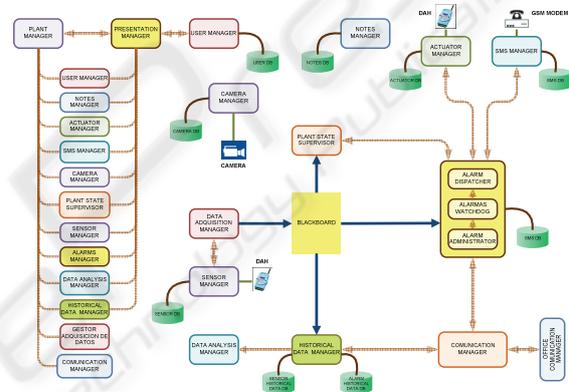


Figure 2: Modular architecture. It can be seen in this picture the different modules of the plant architecture, the interaction with each other and their access to their specific databases.

### 2.2 Data Acquisition Manager

This module takes care of receiving the sensor readings from the sensor manager and transmitting them to the blackboard, from where the other modules take their data as in a consumer/producer classic concurrent problem, which in fact it is. In this case the data acquisition manager will be the producer and the consumers will be the plant state supervisor, the alarm manager and the historical data manager.

Table 1: Consumer/Producer role distribution.

PRODUCER	CONSUMER
Data acquisition manager	Alarm manager
	Historical data manager
	Plant state supervisor

### 2.3 Sensor Manager

This module is in charge of introducing, updating and eliminating the sensors from the system; so this module needs to have access to the Data Acquisition Hardware (DAH). The DAH carries out the signals from the sensors to the final computer. Then these signals can be processed by the computer in order to get the readings in real time.

### 2.4 Actuator Manager

The actuation control is done by this module; it will take care of including, modifying and deleting the actuators from the system, and it will also define in which entry of the DAH it is connected. This module offers an interface so that the user or the rules can activate the different actuators that are installed in the system.

The actuator manager offers digital and analogical outputs in order to control different kinds of switches, motors, etc. This is the part of the system that will enable us with the capacity of automation.

The actuator manager offers us the ability of activating and deactivating the different components of each plant over the distance. We can connect ourselves to the plant via web or via mobile phone and give orders to it. This will prove useful in various situations like shutting down a plant if desired, or a part of a plant (e.g. Water pumps), or activating a plant that was idle.

### 2.5 Alarm Manager

This is one of the most important parts of the system because it is here where we will be able to define the behavior of the control that we want for our plants. With the intention of creating a system capable of giving a solution to a range of situations, as wide as possible, we couldn't define a specific control; therefore, we decided to create an engine so the user could define its own control.

We decided to use a rule-based-system (RBS) where the user would define his own rule set. Besides, we also had to make a special effort to create a solution that could be used by users lacking any knowledge of algorithms; this was achieved by creating an interface that would guide the user through, asking questions and offering different possibilities, and translating them into an algorithm.

Each alarm can have as many conditions for activation as desired. These conditions can be: a sensor's readings are over or under a certain value, this value can either be a constant or another sensor, or how long the condition has to be active to raise the alarm. It is also possible to include a difference between two different sensors, controlling these way possible undesired fluctuations.

The alarm manager is composed of three subsystems which deliver complementary but different functionalities. The subsystems that compose the alarm manager are the following ones:

- Alarm administrator. This subsystem takes care of creating, modifying and deleting the rules from the system. This will be the subsystem that takes care of the alarm database in which the characteristics, conditions and actions will be stored.
- Alarm watchdog. The watchdog evaluates the activation conditions of the rules in order to detect the activation and deactivation of each and every one of them; when it detects an event it informs of it to the alarm dispatcher.
- Alarm dispatcher. This subsystem will give instructions to other modules in order for the actions to take place; such actions can involve sending an SMS to a user to inform of the alarm, activating an actuator so that it can put the plant back in order, or both.

### 2.6 SMS Manager

This component introduces the functionality of sending SMS to the users so they can know in every moment if something is going wrong, and if it comes to that, they can send an SMS to the plant to activate or deactivate any actuators. This way we introduce a lot of possibilities of remote controlling the plant.

The SMS sent by the user must be written according to the specifications of the system so the SMS manager can interpret the object of its action. We have chosen to make the remote control over mobile phone because it enables us to communicate in places where an internet connection might not be possible and because it is a cheap technology.

### 3 REAL EXAMPLES OF INDUSTRIAL PLANTS ALREADY USING THE SYSTEM

We are going to outline several examples where the system is already in use, in order to obtain a real source of data that will help us to decide whether the goals of the project have been achieved, or whether changes have to be made.

#### 3.1 Desalination Plant

Several desalination plants are working with this system at the moment. There is a company that builds desalination plants offering this system to all their new clients, so they are selling not only the final product, the desalination plant, but also its maintenance. The principal use of the system in this case is to control pressures, flow and temperatures of the water during the process of water desalination; it has been introduced in different kinds of contexts as airports, golf courses, etc. They have found that, because of the remote locations of most plants, their capability of detecting problems and failures has been increased substantially. Now they can make a much more efficient use of their human resources and they can give support to their clients in a shorter period of time.

#### 3.2 Student Residence

Due to the request of the manager of the residential service of the University of Las Palmas de Gran Canaria for modernizing the installations of the different buildings belonging to the university, the system was offered as an alternative which has been accepted by the manager as a quite promising possibility.

In the different buildings the control required was mainly of temperature, flow level of water tanks, actuation over the water pumps and the valves, control over the gas tanks, etc. it was also a project that needed a system able to control buildings that were situated in different parts of the city.

At the moment this project has been installed and it is facing different tests of performance, in order to know whether more sensors or actuators are required and if any adjustments are (required) to be implemented to the installations.

### 4 CONCLUSIONS AND FUTURE WORK

As we have observed by the feedback of the final users of the system and our studies of the real cases, we think that the system has developed into broad use software capable of adapting to the needs of most of the industrial plants we have dealt with. This is also because the project has been developed in two phases; in the first one the system had less features and possibilities of control, this was the system placed at the desalination plants, whereas in the second phase the user's requirements were brought into consideration.

As result of the second phase the actual system was developed as an evolution adding the features that were suggested by the user and thought through by us. As future work further versions of the system could be made, for example an expert system could be integrated as an alternative to rule based system, in the same way further features of telecommunication could be included in order to enhance communication with the plant from places as remote as possible, etc.

In conclusion, this system will give answer to systems that will need control and automation for industrial plants without the need to create specific software to control it and with numerous features, such as control over mobile phone, a rule system that automates processes, etc, and that will give answer to the possible needs that it could encounter.

### REFERENCES

- Deepak Alur, John Crupi, Dan Walks. *Core j2ee patterns. Best practices and design strategies*. Sun Microsystems.  
 Nadir Gulzar. *Practical j2ee application architecture*. Mc Graw Hill, Osborne.  
 Chuck Cavaness. *Jakarta Struts*. O'REILLY.  
 Rima Patel Sriganesh, Gerald Brose, Micah Silverman. *Mastering Enterprise JavaBeans 3.0*. Wiley Publishing.  
 Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. Mc Graw Hill.  
 Ian Sommerville. *Software engineering*. Addison Wesley.  
 Solinis Camalich I., Quesada Arencibia A., Rodríguez Rodríguez J. C., Moreno Díaz jr R. An open modular system for monitoring remote sensors. *EUROCAST 2007. Lecture Notes in computer science Vol-4739*, pp: 82-89. 2007.