Intelligent multi-agent architecture for a supervisor of a water treatment plant.

E. Mendoza¹, J. Andramuño¹, J. Núñez², L. Córdova¹

¹Automation Department. Universidad Católica de Santiago de Guayaquil, Ecuador
²Department of Energy. Universidad de la Costa, Colombia

*Corresponding author: eduardo.mendoza01@cu.ucsg.edu.ec

Abstract. The rapid development of Information and Communication Technologies (ICT) and high-capacity hardware components make it necessary to achieve a strong integration of automatic systems based on new paradigms on intelligent distributed architectures, where require highly complex supervision and control tasks, due to the generated requirements of the new production systems, the high number of variables to control and the advancement of technologies, especially in industries where continuous processes have been established. In the present work, a distributed hierarchical modular architecture is proposed for a supervision system, based on multi-agent systems (MAS), oriented to the management of processes in the filtration stage of a water purification plant, using a methodology to the implementation of intelligent agents that allow to project, design, verify and validate the system. This methodology is fundamentally based on the use of the Unified Modeling Language (UML) for its projection and Petri nets (PN) for the simulation and validation of properties, which allows to guarantee the modularity, flexibility, and robustness of the proposed system. The architectures of the intelligent agents in the different programmable devices are modeled and simulated to achieve an adequate interaction and collaboration, allowing to reduce the conflicts that may be generated between them. The evaluation of the distributed architecture focuses on the fulfillment of the functional requirements and evaluation metrics, which, through the analysis of the properties of the Petri net, allows to determine the correct operation of the system and its dynamic behavior in the face of unforeseen situations at different levels of automation.

Keywords. Multi-agent systems, automation systems, water treatment plant, UML, Petri net.

1. Introduction

At the present time, industrial processes face great challenges to achieve high levels of production with the quality of the product demanded by the market, requiring for this to maintain high levels of safety, reliability and availability of the processes and the systems involved [1, 2, 3]. The development of new technologies has made possible the integration of industrial process data in a distributed way, leaving behind the classic automation systems based on local information, until reaching applications based on web services that achieve the integration of information and the data distributed at the different levels in the industrial supervision systems of a company [4, 5, 6]. The need for integration of automation systems (information systems + decision-making systems + decision-making support systems) generates automation that is complex to implement, especially if there are no adequate platforms and control elements, both decision-making and regulatory, which allow the achievement of integrated automation.
of production operations [7]. In this sense, an important role is played by the technology of Multi-agent Systems (SMA) in industrial automation environments [8, 9, 10].

According to [4], SMA technology constitutes an alternative to introduce the new information and communication paradigms in industrial automation by virtue of its properties of autonomy, communication, reactivity, intelligence, and mobility, among others [11, 12]. In addition, they meet three fundamental characteristics in industrial environments seen from agent-oriented modeling, which are the real-time requirements, the generation of knowledge, and the possibility of solving the heterogeneity of the systems, which must be integrated into the forms of articulation. community by agents [13, 14]. The SMA can be applied in dissimilar production and service processes, including water purification plants, in which, despite the economic-social importance, no reports of the application of the SMA have been found in the reviewed references. One of the fundamental processes of automation in water treatment plants is centered on the filter subsystem, in which there are reports of automation solutions based on traditional techniques [15, 16, 17, 18, 19, 20]. In this work, the proposed collaboration models and protocols are proposed and evaluated to build an intelligent distributed architecture that can be considered as an SMA, to perform an analysis of functions and the verification of properties using the Unified Modeling Language (UML) and Petri nets [20, 21]. For this, the study and analysis of the automation process of the filters of a water purification plant and the design of a monitoring system to evaluate its application will be used as a starting point.

2. Analysis of the General Automation System

Drinking water treatment plants can use different technologies to achieve the same goal, but basically, they follow a structure where the main threads that are executed consider coagulation, flocculation, sedimentation, filtration, and the chlorination process, which constitute the fundamental cycle of the water to be treated [22, 23]. This work focuses on the filtering thread, which has sufficient complexity for the application of an SMA methodology.

2.1 Analysis for Filtering Thread Automation

Filtration is a water purification sub-process in which the particles and microorganisms that have not been retained in the previous treatment sub-processes are separated, which is why filters are the most important equipment of the water treatment plant and its operation is highly dependent on the quality of the water as a final product. During the useful life of the filters, they are subjected to several backwashing or filter washing processes, because as the pores of the filter sand become dirty, an increase in the pressure drop occurs until reaching a value maximum that is given by the height of water above the sand of the filter. The duration of this washing process (known as the filtration run) is the working time of the filter between washing and can range from 12 hours to several days.

Filter washing is the operation by which the filtration process is suspended, and water is injected through the bottom of the filter (drains) with adequate pressure, with the objective that the filter bed expands, the grains are rubbed and all the material that has remained between them in the filtering operation is detached. The water treatment plant that is proposed as a case study in this work has a battery of fast and open filters with six independent filtration cells, as shown in figure 1. To maintain the minimum capacity of the plant, the operation is required. Simultaneous filtering of at least four filter cells.

The arrival of the settled water to the filter bank is carried out through the interconnection tube with the settler from its bifurcation, each filtration unit consists of five valves with different functions, figure 1, water inlet to be filtered (V1), filtered water outlet (V2), backwash water inlet (V3), wastewater outlet (V4) and drain outlet (V5). To enable washing, a level greater than the minimum of 0.2 m in the storage tank will be necessary. For lower values, the filter washing process will be blocked, although the level can be adjusted during the operation of the system.

Next, general aspects are described to develop the automation process of the six filtration cells. In the automation of a water treatment plant, the subsystems corresponding to the control and supervision of the operation of the filters would be integrated with the rest of the subsystems.
of the plant. For processes of this type, various automation solutions have been reported [24, 25, 26].

![Filter subsystem in a large capacity water purification plant](image)

**Figure 1.** Filter subsystem in a large capacity water purification plant

The main functions and requirements of the filter automation system that have been considered are the following:

- Measure the level in the filter.
- Measure the water pressure at the washing inlet on the filter.
- Check the operating states of the filter (filtration, washing, rinsing, or draining).
- Check the level in the filter during the filtration state.
- Check the inlet pressure of the filter washing water.
- Check the water level in the wash tank.
- Check the start / stop sequence of pumps B1 and B2 for filling the wash tank.
- Acquisition and registration of all the variables in the six filtration units, the storage tank, and the washing tank.
- Generation of alarms for abnormal states of the variables, establishing priority levels as required.
- User interface with restricted access to configure the system, display, or change the desired values of the controlled variables.
- Failure diagnosis of all sensors, actuators, and pumps of the six filtration units, the storage tank, and the washing tank.
- Fault tolerance functions in the control of the water pumps for the wash tank.
- Production management of the filters to maintain the minimum production capacity required in the plant (keep four filters in a state of filtration, manage the washing process and consider the diagnosis of equipment failures).
- Maintenance management based on the diagnosis of all the sensors, actuators, and pumps of the six filtration units, the storage tank, and the washing tank.
- Remote access via mobile device, limited only to the supervisor's viewing.
- All signals will be displayed on the supervisor's screen. The representation of the filtration units, the storage tank and the washing tank must guarantee real-time visualization of the operating status of each subsystem.

The sequence of filter operations depends on the four possible states, filtration, washing, rinsing, and draining. In addition, the water pressure for washing the filters and the filling process of the washing
tank must be controlled. All subsystems must be controlled by configurations of the Programmable Logic Controller (PLC) to execute the respective sequences of operations.

3. Modular structure of automation for the case study

Based on the functional requirements of the industrial automation process under study, a hierarchical architecture illustrated in Figure 2 is proposed. To guarantee distributed intelligence, a modular design of the architecture is necessary, in such a way that the existence of communication networks from the sensor level to the supervision level. The proposal of physical elements with these characteristics constitutes the basis that should allow the two-way exchange of data to process and transform them into useful information that helps in decision-making. To satisfy the requirement for intelligent field instrumentation, the Modbus RTU network [20, 27] is proposed, which offers the possibility of transmitting standard data and safety-oriented data over the same cable.

Other research works propose the use of sensors with microcontrollers to manage intelligence in these devices in such a way that a more robust and efficient automation system is achieved [24, 25, 26, 27]. In our research, it is proposed to use SIEMENS SIMATIC S7-1200 series compact design controllers, since they have a high integration scale, great real-time calculation capacity and powerful communication possibilities that include an Ethernet interface. Regarding the supervision system, the use of 7-inch KINCO-brand panels with Human-Machine Interface (HMI) is proposed, which have sufficient features, present a powerful development environment, and have an adequate cost. One of the HMIs will preferably be used to visualize all the variables and the dynamic behavior of the filters and another for the cistern and washing tank subsystem. In the intelligent supervisory system, for the development of advanced automation features such as production and maintenance management, it is proposed to use the SCADA CODESYS HMI SL from a PC with wireless connectivity to the automated system.

In figure 2, the agents defined in the levels of instrumentation, control and intelligent supervision are observed. The architecture was made from the description made in the previous section to establish the functions and tasks of each of the agents corresponding to each level. Conflicts and collaborations
between intelligent agents are determined, which guarantees the proper functioning of the system. The internal interactions within the agent are called internal conflicts or internal collaborations and the external conflicts or external collaborations [28, 29, 30].

In general, the advantages offered by SMAs for intelligent integrated automation can be very significant, however, from the perspective of designers there is an increase in the complexity of the design [5, 31, 32], which constitutes a challenge for developers of design, modeling and simulation tools, as they must also take into account the possibility of new phases during the analysis and design stages of automation projects. An alternative for its solution can be given in the methodology suggested in figure 3 [3].

![Figure 3](image_url)

**Figure 3.** Stages of a development methodology for SMAs

Considering the architecture proposed in figure 2, the stages that the methodology of figure 3 must meet for the case study are analyzed below.

3.1. Design and Analysis

For the design of the intelligent integrated system, the UML language given in [33] and the SMA methodology detailed in [4] are used. Based on these results, the design of an architecture of distributed intelligent agents is proposed that allows evaluating their cooperation and conflict resolution, complying with the local and general requirements of the process with the greatest efficiency. The UML modeling tools, specifically the use case diagrams, allow evaluating the functionalities of each of the components and their interrelation for the fulfillment of the general objectives. For this reason, an initial phase of "definition" is established in Figure 2.

3.2. Simulation and Modeling

The structure and operation are simulated in the Simulink software and the hierarchical Petri nets extended in the Matlab and Visual Object Net programs. In the second stage of the proposed design, the modeling, simulation, verification, and validation of the industrial automation models are carried out. These models are implemented from the use case diagrams using UML diagrams (sequence diagrams, class diagrams and / or state diagrams or models in Petri nets (PN GHENeSys hierarchical extended and / or hybrid). Sequence allows establishing the interrelationships between agents based on a time base. Class diagrams are established for the parts of the system that require programming in superlanguages and a guide for the programs of the local control devices. State machines, allow to establish the fundamental stages that systems fulfill for the fulfillment of their functions, and which are the indicators for state changes. However, any of the UML diagrams represent the behavior of the system statically. For this reason, the translation of state diagrams to Petri nets allows the simulation of the dynamic behavior of the system [34].

Verification of the model properties makes it possible to determine some important characteristics, such as the liveliness or absence of blockages, which makes it possible to guarantee that the application programs created from said models in PN are also absent from unplanned stops. The validation by
dynamic simulation of the model makes it possible to guarantee that the programs fulfill the required functions.

3.3. Programming and simulation of industrial or service applications
The third stage of the integrated automation system considers the programming of the application by translating the verified and validated models into an application program package. The programming is initially carried out on program design tools that allow validating the applications created, since they are tools that allow simulating the behavior of the programs within the local control devices, the windows and programs of the supervision and the communication networks, using various tools, for example, the CoDeSys software. This helps reduce programming errors before implementation.

3.4. Implementation of Integrated Automation
In the last stage of development of the intelligent integrated automation system is the study of the particularities of the technology to be used to adapt the programs and models already validated to a technological architecture of application to the filtering process of the water treatment plant. In this stage, the coordination of activities in real time of each function of the transactional domain is carried out, in which tests are carried out at an experimental level to validate the results with comparison metrics that are internationally recognized.

Next, a procedure is presented with the methodology proposed for the formal description of the SMA proposed in the automation of the filters of the water purification plant.

4. Method for the formal description of the proposed Multi-agent System
The studies and tests carried out allow the application of the SMA design method proposed in our research, which consists of six stages to achieve the expected results. The stages or steps of the proposed SMA are the formal description of the agents for each of the agents, the use case diagrams of each level of abstraction, the class diagram of each level of abstraction, the sequence diagram of each level, the state diagram of each type of agent and finally the Petri net of each type of agent. For reasons of extension of these activities, only the analysis of the Level Sensor deliberative Subagent is presented, analyzed through the six steps.

4.1 Step 1
The architecture or model of agents Beliefs, Desires, and Intentions (BDI) is associated with beliefs, desires, or objectives to be achieved and intentions or plans to act and achieve those desires [35, 36]. In formal terms, this logic can be used to describe the components or agents of the SMA. In the practical order, beliefs can be seen as the state of the world. Desires or goals can be associated with a utility value so that desires can be differentiated, as these can be valued using "path formulas" by which all possible paths associated with a desire can be evaluated. Also, in practice, assessment functions can be used to dynamically update goals. On the other hand, the intentions reflect the actions that must be taken to achieve the objectives, so the intentions indicate the actions along a path formula in the decision tree used to calculate the goal values [33].

The BDI architecture analyzes the system as a rational agent with certain mental attitudes present in human beings, emphasizing beliefs, desires and intentions, which represent the information (Beliefs), motivation (Desires) and deliberative (Intentions) states. ) from the people. These attitudes of BDI agents, unlike reactive agents, allow a proactive behavior of the system to try to fulfill its own plans and objectives, which is of vital importance to achieve better performance.

The basic model of the BDI architecture presented in figure 4 proposes the following behavior scheme for the agents [37]:
- Perceive changes in the environment in which it operates.
- Review your beliefs about the world based on your perception.
- Reason about your intentions to reconsider them if necessary.
- Select an action, reasoning about your beliefs and your intentions.
• Execute the selected action, which produces changes in the environment, and return to step 1 to perceive the changes that have occurred.

![Figure 4. Basic BDI Architecture Outline](image)

Next, the table with the beliefs, desires and intention of the Filter Level Sensor Subagent is shown considering the BDI model based on agents sensitive to individual behavior. The Production Planning Agent, Filter Agent, Filtration Controller Subagent, and Actuator Subagent V1 have also been developed, but due to space limitations they have not been included in this work.

| Table 1: Beliefs, desires, and intentions of the Subagent Level Sensor Filter |
|---------------------------------------------------------------|
| **Beliefs**                                                                 |
| Level: Smart Instrumentation                                  |
| Real level: Real level of water in the filter (m)              |
| Measuring range: Minimum and maximum values of the level in the filter (m) |
| Sampling time: (s)                                             |
| Maximum speed of level variation in time: (m / s)              |
| **Desires**                                                   |
| Perform level measurement, processing and analyzing data to validate, condition and detect measurement failures |
| **Intentions**                                                |
| Periodically measure filter level sensor signal               |
| Store the values of the last 10 measurements                  |
| Measured value in measuring range and out of measuring range  |
| Variation rate of measured value OK                           |
| Variation rate of measured value out of range                 |
| Conditioning measurement value                                |
| Average sensor output values with the last 10 measurements    |
| Sensor working OK, Sensor working abnormally                  |
| Faulty sensor; Calibrate sensor; Replace Sensor               |

4.2. **Step 2**

The UML modeling language is the most widely used standard for specifying and documenting any system precisely [33]. The use of UML as a tool for system modeling is an important aid in describing
the functional requirements of the system [21]. Figure 5 shows the diagram of use cases corresponding to the internal architecture proposed for the Subagent Agent Level Filter Sensor.

Figure 5. Detailed Architecture of the Deliberative Subagent Level Sensor

Figure 6 shows the diagram of use cases of the intelligent supervision abstraction level, where agents have been included as special actors to obtain an overview of the system, identifying all the functionalities present in the system and the interactions between the agents of this level. This diagram shows the interaction between agent and subagents that represent the levels of supervision, control, and intelligent instrumentation, within the respective subsystems modeled for each use case.

Figure 6. Diagram of the use case of the Agents of the intelligent supervision level

4.3 Step 3
The class diagrams of each level of abstraction represented by the use cases analyzed in the previous section are presented to model the static structure of the system, identifying the relationships between the classes, their attributes, and methods, including the characteristics of the SMA [33]. Figure 7 shows the agent class diagram for the smart monitoring level. In it, it can be noted that the Intelligent Supervision Management Agent is a fundamental virtual actor of this level, in fact, through its own subagent interface, the interaction of the SMA with the real actors of the case study takes place.
Figure 7. Smart Monitoring Agents level class diagram

4.4 Step 4
Figure 8 represents a Sequence diagram that corresponds to the level of intelligent supervision agents, where all the Agents of the system are related to identify how the groups of objects interact with each other as a function of time, including the exchange of messages.

Figure 8. Smart Monitoring Agents Level Sequence Diagram
4.5 Step 5

In the set of agents that represent the entire SMA, five types of agents were selected that, due to their functions, tasks, and level of abstraction to which they belong, can be considered representative of the rest; the selected agents were the Production Planning Agent, the Filter Agent, the Filtration Controlling Subagent, the Filter Level Sensor Subagent, and the Actuating Subagent. In correspondence with them, the state diagrams of each type of agent are presented to analyze the behavior of a specific object, containing the rules for each transition. Figure 9 shows the state diagram modeling the Filter Level Sensor Deliberative Subagent.

![Figure 9. Deliberative Subagent Status Diagram Level Sensor Filter](image)

5. System modeling and results analysis

After performing the static modeling of the system and its types of agents, a dynamic modeling is carried out using the GHENeSys Petri Network of each type of agent, performing a direct translation of the State Diagrams. Auxiliary places and actions are added to each diagram to provide a non-abstract representation for the most important transitions of the network, allowing the verification of properties of the models (liveliness and reachability), their functional validation and a direct translation for the programs in the IEC61131 [34]. Figure 10 shows the dynamic model in Petri nets that correspond to the state diagram of Figure 9.

![Figure 10. GHENeSys Petri Net Diagram of Sub-Agent Level Sensor Filter](image)
For the validation of Petri nets, the most used techniques are the observation and inspection of a model, the execution by a simulator and the animation and generation of code for its subsequent execution [35]. To validate the proposed models, the behavior and functionality of the models is inspected by running the simulation. The P-invariant equations of each of the elements to be studied are also searched to verify the reachability of the states and the liveliness of the network, which allows to verify that from the initial state any state can be reached avoiding closed loops and blockages; The equations are shown in table 2. Furthermore, it was found that they comply with the property of being persistent since with any two transitions activated, the triggering of one of the transitions does not deactivate the other.

| Table 2. P-invariant equations of the Subagent Level Sensor Filter |
|---------------------------------------------------------------|
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5$ |
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$ |
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5$ |
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5$ |
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5$ |
| $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5$ |

For a more detailed analysis of the proposed networks, in addition to the validation and verification of functional properties, we proceed to verify some of their structural properties such as the presence of place and transition invariants and controllability. To achieve this objective, it is possible to use several methods, including analysis by reduction or transformation, the reachability tree and the Gauss-Jordan elimination tree [36]. For the analysis, the Gauss-Jordan elimination method is used in which, through the calculation of the ranges of the incidence matrix A, the reduced incidence matrix RedA, the transpose of the incidence matrix AT, and from the reduced transposed incidence matrix RedAT, as well as the number of columns of these matrices m, some of their structural properties can be verified. To carry out the calculations, the Matlab® software was used, which facilitated the process of obtaining the matrices A, RedA, AT, RedAT and the rank of each of them. Figure 11 shows the image of these matrices for each of the agents or subagents under analysis.

![Figure 11. Resulting matrices in Matlab of the Subagent Level Sensor Filter](image-url)
Table 3 shows the data of the incidence matrices of the Filter Level Sensor Subagent according to the Gauss-Jordan elimination method, knowing that the network meets the following properties: there are transition invariants, there are place invariants, it is consistent, repetitive, conservative, structurally bounded and not totally controllable.

| Table 3. Data of incidence matrices of the Subagent Level Sensor Filter |
|---------------------------------|------------------|
| # of columns $m$ | Range |
| Incidence Matrix A | 9 | 8 |
| RedA Reduced Incidence Matrix | 9 | 8 |
| AT Transpose Incidence Matrix | 9 | 8 |
| RedAT Reduced Transposed Incidence Matrix | 9 | 8 |

6. Conclusions
The development of this research work allowed modeling the supervisory system for the filtering stage of a water purification plant, considering a multi-agent architecture, where UML and Petri nets have been used as design and analysis tools to determine the static and dynamic behavior of the system. In addition, a modular structure of the integrated automation system was defined, describing the functions associated with the blocks proposed for modeling as a discrete event system.

The proposed modular design, according to the general architecture, complies with the model defined for each of the main functions implemented in the supervisory system, facilitating integration into the general system, and presenting advantages when making modifications without having to alter or stop fully the process. The analysis of the general characteristics and functions of the system and the hardware architecture and the communications network designed allowed to form an SMA architecture that includes instrumentation, control, and intelligent supervision.

The usefulness of UML modeling was proven to specify and document the water filtration system accurately, as well as to describe the scope of the same, the functional requirements, the actors or components, the services or use cases and the relationships between the actors or components. The UML modeling carried out facilitates the process of implementing the automation system at an experimental simulation scale. The models built using Petri nets allowed evaluating the dynamic behavior of the system, checking the details of implementation, algorithms, sequence of operation and the verification of the properties of the subsystems of the distributed intelligent architecture model, which guarantees the good formation of the models and then the dynamic validation of the functional requirements by simulation of the Petri nets.

7. References
[1] Metzger, M., & Polakow, G. 2011. 2011. A survey on applications of agent technology in industrial process control. IEEE Transactions on Industrial Informatics, vol. 7, no. 4, pp. 570-581.
[2] Vilanova, R., Santín, I., & Pedret, C. 2017. Control en estaciones depuradoras de aguas residuales: Estado actual y perspectivas. Revista Iberoamericana de Automática e Informática Industrial RIAI, vol. 14, no. 4, pp. 329-345.
[3] Nuñez Alvarez, J. R., Zamora, Y. P., Pina, I. B., & Angarita, E. N. 2021. Demilitarized network to secure the data stored in industrial networks. International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 1, pp. 611.
[4] Aguilar, J., Bolivar, A.R., Hidrobo, F., & Cerrada, M. 2012. Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial. 2nd Edition, Universidad de Los Andes. Venezuela.
[5] Valdez, J., Pandolfi, D., & Villagra, A. 2018. Red de sensores inteligentes para adquisición de datos de una planta de desalinización de agua. Informes Científicos Técnicos-UNPA, vol. 10, no. 2, pp. 83-95.
[6] Nuñez-Alvarez, J. R., Benitez-Pina, I., & Llosas-Albuerne, Y. 2020. Communications in Flexible Supervisor for Laboratory Research in Renewable Energy. IOP Conf. Series: Materials Science and Engineering, vol. 844, pp. 012016 doi:10.1088/1757-899X/844/1/012016
[7] Michel, F., Ferber, J., & Drogoul, A. 2018. Multi-agent systems and simulation: A survey from the agent community’s perspective. *Multi-Agent Systems*, CRC Press, pp. 17-66.

[8] Oliveira, P., Pinto, T., Morais, H., & Vale, Z. 2012. MASGriP—a multi-agent smart grid simulation platform. *IEEE Power and Energy Society General Meeting*, 2012, pp. 1-8.

[9] Grignard A., Taillandier P., Gaudou B., Vo D.A., Huynh N.Q., & Drogoul A. 2013. GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation. *Lecture Notes in Computer Science*, vol 8291. Springer, Berlin. https://doi.org/10.1007/978-3-642-44927-7_9

[10] Leitão, P., Rodrigues, N., Turrin, C., & Pagani, A. 2015. Multiagent system integrating process and quality control in a factory producing laundry washing machines. *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 879-886.

[11] Dorri, A., Kanhere, S. S., & Jurdak, R. 2018. Multi-agent systems: A survey. *IEEE Access*, vol. 6, pp. 28573-28593.

[12] Jiang, Z., Khalgui, M., Mosbah, O., & Jaouadi, A. 2014. A novel hierarchical multi-agent architecture for automatic restoration of smart grids. *International Journal of Control and Automation*, vol. 7, no. 1, pp. 153-170.

[13] Bravo, C., Aguilar Castro, J., Ríos, A., Aguilar Martin, J., & Rivas, F. 2011. Arquitectura basada en inteligencia artificial distribuida para la gerencia integrada de producción industrial. *Revista iberoamericana de automática e informática industrial*, vol. 8, no. 4, pp. 405-417.

[14] Cárdenas Torres, J. S., & Casas León, J. D. 2017. Diseño de una red hidráulica, automatizada, para la optimización del lavado de filtros de la planta de tratamiento de agua potable de Guasca, Cundinamarca. Thesis, Fundación Universidad de América, Colombia.

[15] Arenas Castaño, F. A., & Londoño Giraldo, W. H. 2017. Diseño de un sistema automatizado para una planta de tratamiento de agua potable. Thesis. Instituto Tecnológico Metropolitano. Colombia.

[16] Patriarca, H A., & Campana, A. 2013. Automatización de acueducto con tecnología GSM/GPRS: Uso de Herramientas Grafcet y GEEMA Aplicación sobre tecnología PLC-HMI-SCADA. Editorial Académica Española, pp. 160.

[17] Calderón Córdova, C., et al., Monitoring and automation of the water pumping, and storage process applied to a water treatment plant. *13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1-6. doi: 10.23919/CISTI.2018.8399292

[18] González-Salcedo, L. O., & García-Nuñez, J. B. 2020. Elaboración de un modelo neuronal artificial para la estimación de turbidez y disposición de dosificaciones en el tratamiento de aguas residuales de la industria avícola. *Informador Técnico*, vol. 84, no. 1, pp. 3-17.

[19] Mendoza, E., Andramuño, J., Núñez, J., & Benítez, I. 2021. Deliberative architecture for smart sensors in the filtering operation of a water purification plant. *J. Phys.: Conf. Ser.*, vol. 1730, no. 1, pp. 012088. doi: 10.1088/1742-6596/1730/1/012088.

[20] Bazy'dlo, G., Wojnawski, M., & Wiśniewski, R. The use of UML and Petri net for graphic specification of the reconfigurable logic controllers. *AIP Conference Proceedings*, 2018, vol. 2040, no. 1, pp. 080004.

[21] Vani, M., Kumari, M. C., Priya, M. H., & Harika, N. 2015. An effective language for object-oriented design-uml (unified modeling language). *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 5, pp. 1212-1218.

[22] Hendricks, D. W. 2006. *Water treatment unit processes: physical and chemical*. 1st Edition. CRC press, pp. 1266.

[23] Faust, S. D., & Aly, O. M. 1996. *Chemistry of water treatment*. 1st Edition. CRC press, pp. 600.

[24] Mendoza, E. V., Fuentes, P., Benítez, I., Reina, D., & Núñez, J. 2020. Red de sensores inalámbricos multisalto para sistemas domóticos de bajo costo y área extendida. *Revista Iberoamericana de Automática e Informática industrial*, vol. 17, no. 4, pp. 412-423. doi: 10.4995/raia.2020.12301.

[25] Andramuño, J., Mendoza, E., Núñez, J., & Liger, E. 2021. Intelligent distributed module for local control of lighting and electrical outlets in a home. *J. Phys.: Conf. Ser.*, vol. 1730, no. 1, pp. 012001. doi: 10.1088/1742-6596/1730/1/012001.
[26] Nuñez, J. R., et al., 2019. Tools for the Implementation of a SCADA System in a Desalination Process. *IEEE Latin America Transactions*, vol. 17, no. 11, pp. 1858-1864, DOI: 10.1109/TLA.2019.8986424

[27] Mendoza, E., Andramuño, J., & Córdova, L. 2020. Intelligent flow and level sensors for a filtering system in a water treatment plant. *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology*, doi: 10.18687/LACCEI2020.1.1.424.

[28] Zarandi, M. F., & Azad, F. K. 2013. A type 2 fuzzy multi agent-based system for scheduling of steel production. *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pp. 992-996.

[29] Fazel Zarandi, M. H., Kashani Azad, F., & Karimi Kashani, H. A. 2017. A Hybrid Modeling for Continuous Casting Scheduling Problem. *AUT Journal of Modeling and Simulation*, vol. 49, no. 2, pp. 173-180.

[30] Taboun, M. S., & Brennan, R. W. 2017. An embedded multi-agent system based industrial wireless sensor network. *Sensors*, vol. 17, no. 9, pp. 2112.

[31] Turcu, C., Turcu, C., & Gaitan, V. 2018. An Internet of Things oriented approach for water utility monitoring and control. *Advances in Computer Science*, arXiv:1811.12807, pp. 175-180.

[32] Legien, G., et al. 2017. Agent-based decision support system for technology recommendation. *Procedia Computer Science*, vol. 108, pp. 897-906.

[33] Zhao, Z., & Xu, Y. 2010. DPMAS: A Design Method for Multi-agent System Using Agent UML. *2010 Third International Conference on Information and Computing*, pp. 137-140, doi: 10.1109/ICIC.2010.305.

[34] Silva, J. R., Benitez, I., Villafruela, L., Gomis, O., & Sudrià, A. 2008. Modeling extended Petri nets compatible with GHENeSys IEC61131 for industrial automation. *The International Journal of Advanced Manufacturing Technology*, vol. 36, no. 11-12, pp. 1180-1190.

[35] Girault, C., & Valk, R. 2013. *Petri nets for systems engineering: a guide to modeling, verification, and applications*. 3rd Edition. Springer Science & Business Media.

[36] Stanimirović, P. S., & Petković, M. D. Gauss–Jordan elimination method for computing outer inverses. *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4667-4679.

[37] García, D. R., Simari, G. R., & García. A. J. 2004. Planificación de agentes BDI. *VI Workshop de Investigadores en Ciencias de la Computación*, 2004, pp. 418-423. [http://sedici.unlp.edu.ar/handle/10915/21240](http://sedici.unlp.edu.ar/handle/10915/21240)