Proceedings in MANUFACTURING SYSTEMS

Proceedings in Manufacturing Systems, Volume 21, Issue 1, 2026, pp. 35-41 ISSN 2067-9238

STRUCTURED TEXT AND INDUSTRIAL INTERNET OF THINGS CONVERGENCE FOR SMART MANUFACTURING

Roxana Florina BOANDĂ[1,*], Aurel FRATU[2]

[1)] PhD, Student, Department of Automation and Information Technology, Transilvania University of Brașov, Brașov, Romania
[2)] Prof. Dr. Habil. Eng., Full Professor, Department of Automation and Information Technology, Transilvania University of Brașov, Brașov, Romania

https://doi.org/10.5281/zenodo.21823766

Abstract: *In the current context, where the concept of Industry 4.0 is gaining momentum, the separation between operational control (OT) and information technology (IT) is becoming decreasingly visible in the automation industry. This paper will explore the compatibility between one of the programming languages of programmable logic controllers (PLC), namely Structured Text (ST), and Industrial Internet of Things (IIoT) ecosystems to improve production and the transition to intelligent processing (Smart Manufacturing). While traditional PLCs were designed as isolated systems, this study proposes a hybrid architecture in which deterministic control logic is augmented with Edge Intelligence capabilities and asynchronous communication via OPC UA. Exploring the modernization of industrial architecture achieved by introducing Digital Twin inside of PLC code, the proposed solution will integrate this through the implementation made with Structured Text. The method involves filtering data from the physical field directly at the source, by implementing the logic for processing and selecting information from controllers and not at the Cloud level, so that redundant information is not transmitted. It is desired that this implementation will reduce data traffic and decrease network latency, but without compromising the accuracy of monitoring or the integrity of the industrial process. Finally, the study substantiates the concept of a resilient factory. By establishing bidirectional and coherent data flows between the Cloud and the PLC, the systems become capable of self-optimization in real time.*

Key words: *Structured Text, IIoT, Smart Manufacturing, PLC, Edge Computing.*

1. INTRODUCTION

In the current industrial environment, paradigm conflicts can be interpreted as the interaction between intercommunicating systems that use different languages. Although the IoT ecosystem is built around asynchronous data flows, with variable latencies and databases distributed in the cloud. On the other hand, with a completely different rhythm, built from cyclic executions with absolute and rigid determinism, the PLC (Programmable Logic Controller), focuses on creating and maintaining process stability and safety [1].

The concept of Hybrid Edge Intelligence is the bridge between the two systems. Through it, a permutation of functions takes place, with the PLC becoming an active information processing node and not just an executor, according to its standard function. In this organization, the PLC, by using the Structured Text (ST) programming language, will become a primary tool for filtering information at the source, in addition to the classic command and control operating mode.

The PLC prepares data for Machine Learning models by using normalization, aggregation, and filtering algorithms implemented directly in the ST code. Edge network calculations are used to extract meaningful features in real time without sending noisy data to the cloud.

2. INDUSTRIAL INTERNET OF THINGS (IIOT)

Industrial Internet of Things (IIoT) is the connection of devices, sensors and processing tools of the OT (Operational Technology) level to industrial applications of energy management and production. Compared to traditional automation, which focuses on local processing and control, IIoT collects and analyzes complex data, resulting in global process improvement.

The fourth industrial revolution (Industry 4.0) has made a great contribution to the evolution of IIoT. This evolution has led to the emergence of interoperable communication protocols and a decrease in the cost of sensors [2].

IIot allows for the bidirectional exchange of information between the IT structure and the inside of the control network (OT), outlining the idea of a "connected factory".

2.1. Difference between IIoT and IoT

The context and technical requirements differentiate between the two concepts, even though both are based on the exchange of data between connected devices. IIoT is used in environments where there are strict safety rules,

[*] Corresponding author: Politehnicii St., No.1, V building, Brașov, Romania,
Tel.: 0711928609,
E-mail address: *iamandiir@gmail.com* (R.-F, Boanda), *fratu@unitbv.ro* (A. Fratu).

while IoT focuses more on user comfort and "best-effort" applications.

The biggest difference is observed in latency and determinism. If in a system using IoT a command delay of 500 ms does not obviously affect the user experience, in IIoT such a delay can lead to mechanical collision of a piece of equipment or to compromising the quality of the final product.

If life cycles and resistance in the execution environment are considered, it could be said that robustness and reliability are different properties between the two systems. IIoT devices are used in environments with vibrations, humidity, dust and electromagnetic interference, and they must operate autonomously.

IIoT cybersecurity is much more complex, as the impact is much greater on such a network than on an IoT device. In this context, the industrial protocols mentioned have advanced encryption, certification and authorization mechanisms directly in the device.

2.2. Internet of Everything

Internet of Everything (IoE) The Internet of Everything (IoE) represents the combination of four elements: objects, processes, data, and people creating the intelligent network of all things. IIoT focuses mainly on the interconnection of sensors and machines, while IoE transforms information into action, improving the way connections are formed for added value for production lines.

These connections are the IoE solution for manufacturing lines based on three major stages: Machine-to-Machine (M2M), Machine-to-People (M2P) and People-to-People (P2P) [3]. In the first stage, it maintains the continuity of equipment communication without human intervention (for example, a conveyor belt adjusts its speed depending on how many parts are assembled by a robot, which counts and transmits the number of parts made). The next step consists of transmitting critical data to operators for rapid decision-making in the event of anomalies. And the last stage is the collaboration between experts, who, through real-time data access, can optimize design or maintenance processes.

This synchronization between the 3 stages ensures that the raw data collected from process sensors is processed and shared with the right people, through the right method, in the time needed for decision-making.

3. SYSTEM ARCHITECTURE

3.1. The evolution of ST towards Smart Manufacturing

A clearer synthesis of the impact of the mentioned technologies in modern automation can be seen in the comparative analysis of the development of the Structured Text standard within the system architecture. Table 1 highlights how the role of data processors for cognitive architecture has evolved from the classical attributions of sequential logic implementation.

Starting from static data structures and polling protocols from the "Legacy" standard (1993/2003), the evolution is analyzed by integrating the "IoT-Ready" paradigm (2023-2026). One of the biggest changes is related to the programming architecture, which allows mapping active devices in software as "Digital twins", directly in the PLC, namely the transition from procedural to object-oriented programming (OOP) [4].

Table 1

Comparative table of ST "Legacy" vs. ST "IoT-Ready" and the impact in Smart manufacturing

(source: own processing)

Components	**Structured Text "Legacy" (1993/2003)**	**Structured Text "IoT-Ready" (2023-2026)**	**Impact on Smart Manufacturing**
Architecture	Procedural	Object Oriented	Digital twin for physical devices
Timing	Local (PLC internal clock)	Global (NTP /PTP)	Time-Series Analysis
Data processing	Static	Dynamic	Data encoding
Maintenance	Reactive	Predictive	Reducing downtime
Communication protocol	Master – Slave	Publish - Subscribe	Bandwidth reduction
Cyber Security	Default	Native	Protecting data integrity
Cloud Connection	Middleware – external gateway	Edge-to-Cloud	Eliminate latency

Regarding the interaction with the Cloud environment, the elimination of the external gateway (Middleware systems) and the introduction of native protocols (OPC UA/MQTT) in the source code, the PLC interacts with hyperscale platforms (Azure, AWS) with lower latency and hardware costs [5].

Structured Text is now becoming an important component for an industrial environment, providing transparency, traceability and optimization in data processing, everything being synchronized globally through time precision protocols (NTP or PTP).

3.2. Edge-to-Cloud

In the architecture of industrial systems, intermediate hardware is currently used that introduces additional latency, for simplification it is proposed to introduce the communication directly in the PLC. In the Structured Text language, this implementation is done through native libraries that can instantiate OPC UA (Open Platform Communications Unified Architecture) servers or MQTT (Message Queuing Telemetry Transport) clients [4].

In modern models, runtimes are used, based on TwinCat or CODESYS, which use optimized TC/IP stacks, unlike old models where to distribute data the PLC query is done via Modbus through an external gateway. Through a fluid communication with MQTT brokers, the mapping is implemented directly in JSON payloads, but not before defining them in complex data structures using ST.

At the same time, the integration of the OPC UA Pub/Sub standard according to IEC 62541 into the PLC, without the need for a continuous point-to-point instance, allows for deterministic network communication [6]. Since this assembly uses UDP (User Datagram Protocol) protocols, which allow the simultaneous distribution of data sets to multiple receivers, it increases scalability in IIot (Industrial IoT) architecture.

Regarding the security risks of gateways, the ST code can directly manage digital certificates and handshaking. In the PLC development environment, programmers can parameterize communication paths by using function blocks dedicated to connectivity. At the same time, the ST editor allows for real-time monitoring and visualization of data transmission logic in online mode, considerably reducing troubleshooting times. Furthermore, static aggregation and preprocessing by filtering signals, before they are encapsulated in MQTT packets, leads to a reduction in the volume of data in the cloud [7].

In such an integration, "Store and Forward" methods can also be used in the PLC's memory, to protect data in case of loss of internet connection. By eliminating intermediary systems (Windows, Linux), the attack surface is minimized, increasing cyber security. Thus, the PLC, because it can automatically decide what information should be sent to the ERP/MES, becomes an "Edge Intelligence" node.

3.2.1. Edge-to-Cloud Architecture Diagram

To visualize the data collection and transmission levels of the "Edge-to-Cloud" architecture, three vertical levels are highlighted, which resolve the "paradigm conflict" (see Fig. 1).

The proposed solution contains the following three components:

- Deterministic level (Edge):
  - Piloting Level: where the basic logic runs in millisecond cycles.
  - Hybrid Edge Logic: Pre-processing data from sensors or devices with parameter extraction for AI, obtained through ST code.
  - Stacks: These native communication stacks are directly implemented in the PLC runtime.
- Communication channel
  - It does not use a physical gateway; the PLC communicates directly through protocols such as MQTT and OPC UA.
- Asynchronous level (Cloud & Enterprise)
  - The results of training AI models using historical data are sent back to the edge so that they do not negatively influence the real-time control part.

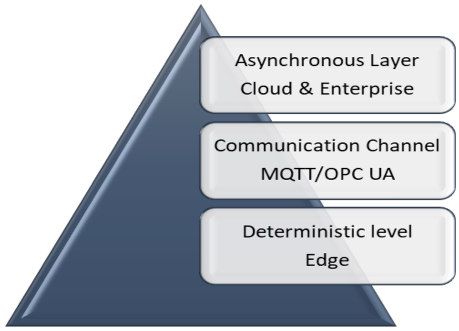

Fig. 1. Edge-to-Cloud architecture diagram for data collection and transmission (source: own processing).

4. MODULARIZING ST CODE FOR IOT SCALABILITY

Current control systems use ST according to the IEC 61131-3 standard, as a transition from operational programming to Object-Oriented Programming (OOP). By eliminating redundancy in highly complex designs, this conceptual framework allows for the incorporation of both IoT metadata and control logic into separate software components. Developers create reusable, extensible function blocks and classes that resemble code modules for much easier maintenance. Thus, IoT scalability becomes an intrinsic element of the code and not an externally added part through software patches.

4.1. Implementing "Digital Twins" at the Controller level

The paradigm shift brought by the development of Digital Twin directly into the controller program in the industrial environment moves complex processing from the Cloud to the deterministic Edge execution. This transforms from a graphical representation, a functional software model developed in the PLC Hardware, running in parallel with the physical process. Therefore, this physical proximity achieves real-time synchronization and transposes the system state at each execution cycle, without the need to consider network latencies. The modularity of the ST programming language supports the scalability of this model. The use of predictive models and the incorporation of control logic into reusable function blocks (FBs) lead to the creation of virtual libraries, quickly instantiated, ensuring the rigorous stability that the automation industry requires.

Simulated, the entire virtual system, separated from the execution logic, the code as well as the structure becomes flexible acting as a pre-flight check validation model. Memory consumption is optimized by using data structures and pointers and allows running, without affecting priority tasks, multiple instances [6].

The concept of Digital Twin can be implemented directly in PLC memory, by using Interfaces, Properties and Methods, which until now have been available only for Cloud platforms. The interface takes care of communication between the various equipment from different manufacturers, ensuring interoperability. At the operational level, the Digital Twin in the controller becomes an abstracted layer of the hardware (HAL), making the transition from raw data collected from sensors or actuators into structured data objects, ready to be transmitted further to processing applications. The cloud, receiving pre-processed data or data about anomalies, can facilitate predictive maintenance by automatically detecting performance variations.

5. TEST PRODUCTION LINE USING PLC SENSOR AND ACTUATOR STATE MONITORING

Monitoring and testing production lines is the core of a robust industrial architecture that transitions from basic automation to complex data analysis. The programmable logic controller becomes a processing node that continuously extracts information from sensor and actuator data and verifies signal integrity, going beyond its basic role of executing command sequences.

5.1. Framework for Real-Time State Acquisition

The software entity created by mirroring physical behavior, using the structural capabilities of the Structured Text language in the controller code, can analyze the system in real time. Thus, the programmer can use structured data types, grouping inputs and outputs with essential metadata to determine changes in the system state over time or the quality of the signals it receives from the physical area. Continuous sensor-level monitoring is used to detect mechanical vibrations and electrical noise-chattering phenomenon on manufacturing lines to prevent unjustified stops or false alarms. On the other hand, actuator monitoring used to detect discrepancies between the command in the code and the mechanical movement, is achieved with a deterministic comparison between the feedback received from limiters or proximity sensors and the command signal emitted by the code.

5.2. Developing "Health Objects" in Structured Text

"Health Object" is a concept of an autonomous software model that makes it possible to track the performance of each industrial asset, by transforming the initial binary signals into monitoring indicators. Implemented as a specialized block function, in ST language, it contains code that implements the monitoring and diagnostic logic [5]. Because it is executed in the PLC run cycle, the time control is precise, because the duration between the activation of the command and the signal received from the feedback sensor can be precisely timed. This basic function is used to identify pressure losses or slowdown of movements caused by excessive friction, for tracking performance over time.

The watchdog function is an essential element in ST logic for monitoring sensor activity. The function block is designed to generate alarms with the sensor and the error code, if this behavior is not expected, within a predefined interval (milliseconds or seconds depending on the nature of the process). Thus, the intervention is made on the faulty component, eliminating ambiguity in diagnostics. The use of such objects facilitates the introduction of new devices, without the need to rewrite the control logic, but only instantiate a new function block.

6. IIOT-FOCUSED STRUCTURED TEXT (ST) TO READING VIBRATION SENSOR PARAMETERS - CASE STUDY

This paper proposes the creation of a simulation model of a production system based on the monitoring of actuators and sensors that are carried out using PLC, for better visibility of operations in normal operating mode.

Compared to the classic simulation model, with discrete events, the proposal in this study, streamlines, without requiring detailed knowledge of the system, control logic or physical inspection, by reducing human resources and the time required for data flow analysis.

The constant evolution in the field of automation and the fact that there is an increasing and necessary requirement for the modularization of production systems, the proposed approach aims to improve the simulation model. By digitally representing manufacturing systems in real time, it offers a scalable and practical solution, which becomes the basis for further development in Digital Twin applications.

To simulate the production process in a digital environment, the Delphi programming environment was used as a virtual model for the chosen case study.

The center of this manufacturing system is the PLC, which through control logic processes the inputs and outputs, thus activating the system actuators. At the same time, the structural construction of the PLC makes them suitable for any kind of industrial environment, including those with high temperatures, humidity or vibrations. PLCs receive information from the physical area, such as the states of inductive/capacitive sensors, the pressure values from transducers, analog values from temperature or vibration sensors, etc., and based on these, they make decisions to accustom the execution systems.

6.1. Generation of the virtual production model - experimental study

The objective of the experimental study is to create a production process simulation environment with data obtained during normal operation, for the analysis of the feasibility of the system. At the same time, the aim is to reduce the amount of prior knowledge about the entire packaging line, on-site assessments or major changes in the PLC code written in ST, which controls this process.

The experimental environment was developed using a virtual model (Fig. 2).

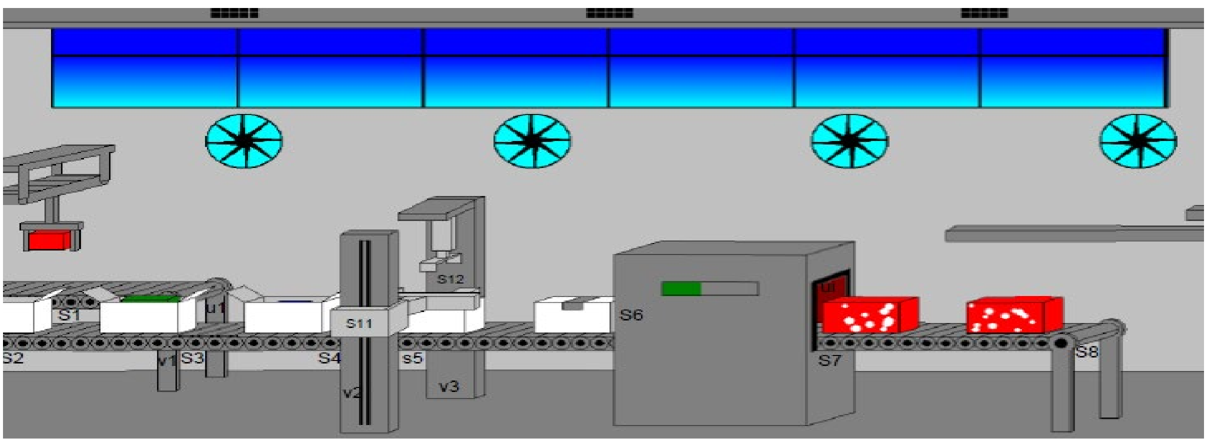

Fig. 2. Multi-stage packaging and labeling line for parcels - virtual model (source: own processing, using Delphi, version 7).

Using Delphi, version 7, the physical components of the system were represented digitally, with the intention that the proposed method be evaluated on production lines with various configurations. This environment offers the virtualization of the physical aspects, without including the control part of the predefined logic deliberately for greater configuration flexibility.

The manufacturing model, taken as an experimental case, consists of a conveyor belt (Fig. 2) synchronized with two independent processing sections – packaging and labeling for packages.

The operating mode of the processing line starts from placing the packages on the conveyor belt that moves in one direction, and when they reach the processing points, the belt stops for processing. For this, each processing section has presence sensors both at the entrance and at the exit, for controlling the conveyor belt

The processing stations also have a presence sensor positioned in the center for process control. The sensors selected for this model are normally closed, having as an input to the plc, a continuous logic high (true) signal in standard conditions.

For product processing, the conveyor belt is stopped to provide greater stability of the line. The initial loading is done once every 45 s, a timing that ensures that only one piece is being processed at a time and the data collected is unique.

In this framework, 8 analog signals and 8 digital signals are defined, corresponding to the monitoring of the operating environment, respectively of the installation sensors. These signals are distributed as follows: s1...s8 – presence sensors, at the entrance, inside, and at the exit of the processing sections, v1...v3 – vibration sensors, and u1 and u2 – humidity sensors.

Communication with IIoT cloud applications is carried out using OPC UA, often found in the concept of Industry 4.0. The server-client architecture is provided by the test environment, implemented between the individual components. For the OPC UA server, the previously developed virtual model (Fig. 2) will be used with the help of Siemens SiOME version 3.2.0. This version serves components made by Siemens, allowing the visual modeling of automation and industrial processes. The graphical interface of this tool offers the user the possibility to interact with object models and modify or create them.

6.2. IIoT-focused Structured Text (ST)

To continue the case study, the processing of the collected data is used using function blocks in ST and a digital twin for engine vibration was created. The results will be mapped directly into the information model created in SiOME, so clients who will connect via OPC UA will access an intelligent data structure according to the IIoT paradigm.

The Digital Twin in the proposed PLC is created separately in a Functional Block (FB) called FB_DigitalTwin (Fig. 3) which will be called in the main block. The function is estimating the vibration deviation between the physical and virtual model, based on the correlation between speed and load. In nominal mode, vibration increases with speed. Thus, if the engine vibrates at 20 mm/s at low speed and the fixed alarm is set to 30 mm/s, it will not be triggered, but the Digital Twin "knows" that at this speed, it should nominally be 15 mm/s, and the deviation will be sent as a predictive maintenance alert. Therefore, the PLC performs raw data analysis at the Edge level and sends only the result to the Cloud, improving the data flow between the two.

For the calculation of RPM feedback (rotations per minute) a sensor was used that detects the presence of a slot on a metal plate mounted on the conveyor motor shaft. To find out the value in real time, a cyclic reading function was implemented every 10 seconds, in which it is read using HSC (high speed counter) of the PLC on which the sensor input is configured. After reading, the counter is reset in the cyclic function. And another calculation function is used to calculate RPM feedback by multiplying the read value by 60.

Next, the scaling, filtering and determination of the system status and Siemens TIA Portal V18 the creation of the data structure that will be sent to the IIot will be done using a function implemented in the PLC (Fig. 4).

```

#VirtualVibrationValue := #BaVibration + (#KFactor * #RPMFeedback);

// LPF  - Low Pass Filter
#FilteredVibrationValue := #FilteredVibrationValue + (#GainFilter * (#VirtualVibrationValue -

// Check difference between DT and value
#InstantDeviationValue := #ActualVibrationMmS - #FilteredVibrationValue;

//Create a threshold value for alarms - settable
#ThreshDeviationValue := #FilteredVibrationValue * 0.3;

IF #InstantDeviationValue > #ThreshDeviationValue THEN
    #HealthStatus := 3; // Equipment NOT healthy
    #AlarmStatus := TRUE;
ELSE
    #HealthStatus := 0; // Sync with DT Model
    #AlarmStatus := FALSE;
END_IF;
```

Fig. 3. Digital Twin Model in PLC - ST Language
(source: own processing, TIA Portal V18 Siemens).

```
// check vibration coditions
FOR #i := 1 TO 3 DO
    IF #OutputDrivesControl[#i]=TRUE THEN
        //transform signal (0-27648) in  (0-50 mm/s)
        #VibrationNodes[#i].Value := (INT_TO_REAL(#VibrationRawVal[#i]) / 27648.0) * #VMax;
        #VibrationNodes[#i].Label := CONCAT(IN1 := 'Sensor_',
                                            IN2 := INT_TO_STRING(#i),
                                            IN3 := '_VIB');
        //call Digital Twin from PLC
        #FB_Dt_proc(BaVibration := #VibrationRawVal[#i],
                    KFactor := 0.30,
                    RPMFeedback := #RPMFeedback[#i],
                    GainFilter := 1.5,
                    ActualVibrationMmS := #VibrationNodes[#i].Value,
                    AlarmStatus => #AlarmDrives[#i],
                    HealthStatus => #VibrationNodes[#i].Status);
    END_IF;
    IF (#OutputDrivesControl[#i] = TRUE AND #AlarmDrives[#i] = TRUE) THEN

        #OutputDrivesControl[#i] := FALSE;
    END_IF;
END_FOR;
// u1..u2 humidity sensors
FOR #j := 1 TO 2 DO
    #HumidityNodes[#j].Value := (INT_TO_REAL(#HumRawVal[#j]) / 27648.0) * #UMax;

    IF #HumidityNodes[#j].Value > 85.0 THEN
        #HumidityNodes[#j].Status := 1; // Warning: High value
    ELSE
        #HumidityNodes[#j].Status := 0;
    END_IF;
END_FOR;
```

Fig. 4. Scaling, filtering and determination of the system status, logic for motor control, and the creation of the data structure in ST Language (source: own processing, TIA Portal V18 Siemens).

The command of the 3 conveyor motors is done in a separate function: the sensors s1 and s2 start by activating the TRUE state of the variable OutputDrivesControl [1] associated with the physical output Q1.0 (Start Motor 1); s3 and s5 start Motor 2 and s7 and s8 start Motor 3.

The FB_ValueProcessing_DT function block was created, having as input data, the presence sensors (s1...s8), the unfiltered value of v1...v3, the humidity value u1, u2, as input-output data the command on the three motors, and as outputs the data structures that will be input data for the SiOME application that communicates with the Cloud.

In the same function, part of the control logic, scaling, information filtering and populating the data structure transmitted to IIoT was centralized. For each vibration value, a signal transformation is made from integer values to real-time values limited to 0–50; the sensor name is prepared by concatenating strings; and for the sensor status, FB_DigitalTwin is called. At the same time, on the production line control side, if there is an alarm after the previous FB is completed, it is decided to stop that motor until its physical verification and confirmation in the HMI of its functionality.

In the case of humidity sensors, the same scaling and filtering is done with UMax = 95; and the status is determined by exceeding the value of 85.0, a fixed value in the program code.

In the SiOME application, the data structures were declared, the project from TIA Portal V18 was imported and the connection between the data times was made, see (Fig. 5).

The configuration presented above facilitates interaction with higher-level systems (predictive analysis platforms, SCADA or Cloud Gateway), by directly accessing the unified VibrationSenzor1 object, without wasting time calling the PLC's internal memory for each component element of the structure.

7. CONCLUSIONS

The integration of Industry 4.0 specific technologies in automated industrial environments, present in this research and in the case study in this paper, demonstrating feasibility and scalability, through a unitary vision of digital transformation. The structured analysis highlighted throughout the paper establishes a link between theoretical foundations and practical development.

The evolution of classical automation towards decentralized industrial systems, presented in the introduction, are the starting point for mapping the current issue, then the correlation with the current state of the recent specialized literature led to the identification of the barrier in existing command and control systems.

To balance the flow of information, the proposed architecture brings intelligent analysis and computing power to the edge of the network through Edge Computing. Comparing the new architectures with Legacy systems, demonstrates the reduction of latencies in the Security decision-making domain, the substantial

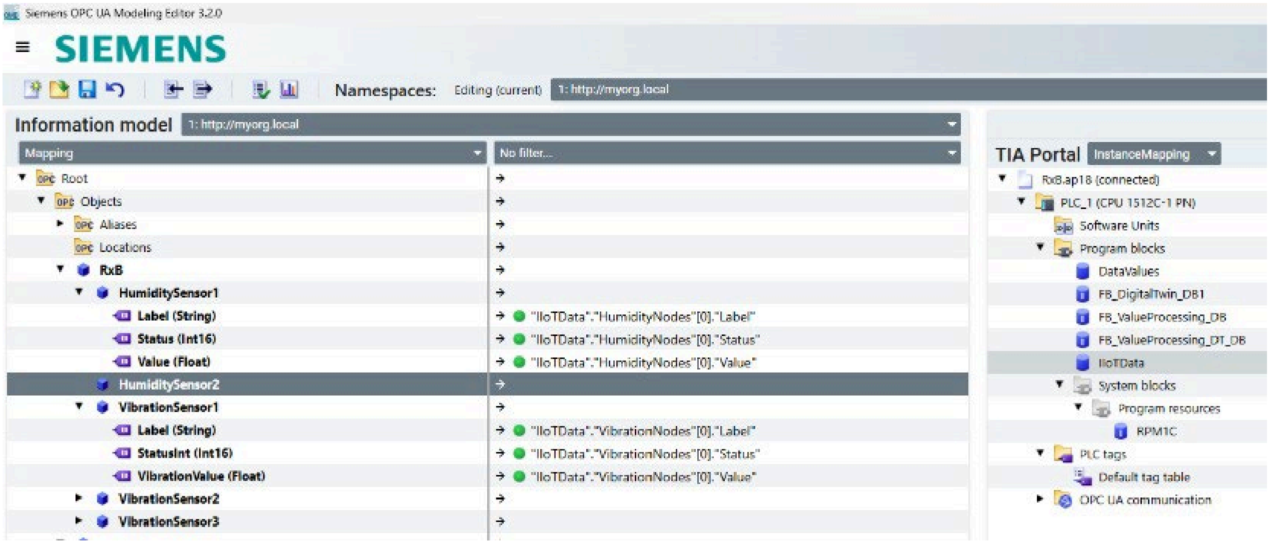

Fig. 5. SiOME interface parameterized for communication with the IIot data structure in the PLC (source: own processing, SiOME V3.2.0 Siemens).

reduction of network bandwidth and the construction of free communication between different devices, through a unified protocol.

This architecture goes from the theoretical to the practical level, with the help of the case study. The implementation in ST language in the TIA Portal environment, of the complex Edge Analytics algorithms, demonstrates that current automation equipment has a real-time analysis capacity. Also, the development of the Digital Twin, at the controller level, makes it possible to move from reactive to predictive maintenance.

Furthermore, by using the SiOME tool, the interoperability problem is also solved. by transforming raw data into digital assets that can be easily interpreted by most Cloud-level systems. Through the OPCUA protocol, the data read in real time is filtered, compared through Digital TWIN, enriched with the time stamp and quality class, creating a compact data flow integrated directly into the PLC, so that Cloud platforms can easily access, through asynchronous and structured queries, unified sensor objects, without the need to use analysis algorithms for data preparation.

In conclusion, this paper demonstrates that the step towards the modernization of intelligent automation starts from the standardization of the way data processing and exposure is done. The structure: ST language with analytical programming, OPC UA communication; SiOME – with semantic modeling; is optimal for a scalable and robust solution of the physical-cybernetic system, aligned with the operational and predictive maintenance objectives of the current industrialization paradigm.

REFERENCES

[1] M. Zheng, S. Wang, J. Li, *Hybrid Intelligence in Industrial Edge Computing: Orchestrating Deterministic Control and AI Inference*, IEEE Transactions on Industrial Informatics, Vol. 18, No. 4, 2022, pp. 2654-2665.

[2] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, *Industrial Internet of Things (IIoT): Resilience, challenges and future perspectives*, Sensors International, Vol. 5, 2024, p. 100231.

[3] C. I. Silvestru, *Internet of Everything (IoE) – The Future of Industrial Connectivity*, Proceedings of the International Conference on Manufacturing Systems (ICMAS), Vol. 16, No. 4, 2021, pp. 157-161.

[4] Y. Liu, H. Zhang, *Deterministic Latency for Asynchronous IoT Services in Industrial Cyber-Physical Systems*, IEEE Transactions on Automation Science and Engineering, Vol. 20, No. 3, 2024, pp. 1540-1555.

[5] L. S. D. Santos et al., *Bridging the Gap Between Deterministic PLC Cycles and Stochastic Machine Learning Models at the Industrial Edge*, IEEE Access, Vol. 10, 2022, pp. 115203-115218.

[6] F. P. Bakhtiari et al., *Evaluation of MQTT and OPC UA Pub/Sub for High-Frequency Data Streaming*, IEEE Open Journal of Control Systems, Vol. 2, 2023, pp. 112-125.