OFFSITE DEVELOPMENT OF MANUFACTURING CONTROL SOLUTIONS WITH DIGITAL TWIN

Andrei-Alexandru CRISTEA[1], Silviu RĂILEANU[2], Theodor BORANGIU[3], Ionuţ Valentin CRISTEA[4]

Abstract. *This article introduces a methodology that leverages the Digital Twin concept to develop, test, and validate advanced control strategies within a virtual environment prior to their onsite implementation. The primary emphasis is on replicating real-time conditions using high-fidelity models and incorporating IoT data streams to accurately reflect actual operations. By integrating digital twins with edge and fog computing architectures, the proposed framework facilitates the remote design of semi-heterarchical control systems, which can be seamlessly incorporated into smart manufacturing settings. The paper examines how this strategy minimizes downtime, expedites deployment, and enhances system reliability while maintaining responsiveness and adaptability in dynamic production environments. The reported research details an architecture based on the open-source event flow programming platform Node-RED. Experimental results are provided for connecting a Digital Twin-based solution for ABB industrial robot programming (RobotStudio) with a unified environment for designing, programming, and managing PLC-based systems from Siemens - the Totally Integrated Automation (TIA) Portal.*

Keywords: digital twin, integration, smart manufacturing, industrial robot, PLC.

DOI 10.56082/annalsarsciinfo.2026.1.19

1 Introduction

The field of digital production is evolving rapidly. Operational velocity, adaptability, and efficiency are now crucial for enterprises aiming to sustain high output amid shifting market demand. In the context of Industry 4.0 [1], defined by digital transformation and intelligent automation, companies face new demands. Manufacturing systems must not only be fast, but also flexible, modular, and resilient to supply chain disruptions or market changes [2]. Meeting these requirements calls for implementing and validating control systems in virtual environments before actual deployment; this approach is especially effective. This

[1]Ph.D. student, POLITEHNICA Bucharest, andrei.cristea2107@upb.ro
[2]Associate Professor, Ph.D., POLITEHNICA Bucharest, silviu.raileanu@upb.ro
[3]University Professor, Ph.D., POLITEHNICA Bucharest, Academy of Romanian Scientists, theodor.borangiu@upb.ro
[4]Ph.D. student, POLITEHNICA Bucharest, ionut.cristea1507@stud.acs.upb.ro

method is especially important when the physical equipment—such as industrial robots, transport lines, packaging machines, or other key production assets—cannot be used during development because they are needed for ongoing operations. Simulations allow developers to identify integration problems early, reduce commissioning time, and provide a safe, non-disruptive way to shift to live production. For this paper, onsite development refers to any actions taken at the client location (e.g., the factory). Offsite development covers the actions to create the control solutions at the contractor (developer) location.

Industrial automation of workstations employing material handling assets—primarily realized through industrial robotics—relies on a variety of distinct control mechanisms, including:

- Robot controllers dedicated to path trajectory automation, as well as managing the operational execution of the terminal end-effector;
- Industrial-grade PCs responsible for running vision-based algorithms for both quality assurance and guiding robotic operations;
- Embedded platforms tasked with decentralized data processing, remote system supervision, and integration with cloud-based infrastructures;
- Programmable logic controllers (PLCs) to regulate material conveyance and presentation, enforce workstation safety protocols (utilizing intelligent doors, laser barriers), and facilitate seamless integration with operational equipment—specifically, robotic manipulators.

The control devices function within a hierarchical architecture, where the PLC assumes the role of the master (central) controller, while the robot and its associated subsystems (inclusive of safety mechanisms) act as subordinate nodes. The implementation of these robotized workstations directly at the client's facility is often challenging, as such projects typically involve retrofitting or upgrading pre-existing, continuously operating solutions (for example, in the automotive sector, where manufacturing persists across three shifts daily for six days each week, reserving the seventh day exclusively for preparatory testing).

This work introduces an approach for unifying control systems associated with equipment sourced from various manufacturers, thereby supporting the virtual commissioning of automation solutions. The proposed method encompasses the integration of digital models and Digital Twin (DT) [3] development platforms—whether for industrial robotics or PLCs—as well as applications such as machine vision simulators and data acquisition tools, in addition to compact, cost-effective embedded devices frequently accessible to developers. Central to this strategy is the adoption of the Manufacturing Service Bus (MSB) [4], an established paradigm within the manufacturing domain.

2 MSB Concept-Based Proposed Architecture

The digital representations described earlier enable both the configuration of industrial equipment and interaction with external software through a variety of services and communication protocols. Nevertheless, certain obstacles persist. A significant issue is the reliance on digital or analog signal-based interconnections, a practice prevalent in contemporary automated workstation setups.

Within this framework, the present paper introduces an approach for linking digital models originating from a variety of manufacturers. A central orchestration entity governs numerous concurrent data streams and adapts them to the specific requirements of each connected system. This paradigm, commonly referenced in academic literature as the Enterprise Service Bus (ESB), constitutes a middleware communication infrastructure for software entities—specifically Digital Twins—built upon Service-Oriented Architecture (SOA) principles. When tailored to industrial automation, this infrastructure is commonly referred to as the Manufacturing Service Bus (MSB), notable for its compatibility with a diverse array of industrial protocols such as Profinet, Open Platform Communications (OPC), BACnet, Modbus, TCP/UDP, among others. Beyond providing seamless interconnection, the approach also introduces a methodology that simplifies the augmentation and extension of control applications developed within the digital model environment. The overall system architecture is depicted in Figure 1, serving as a digital complement to the physical integration of industrial assets.

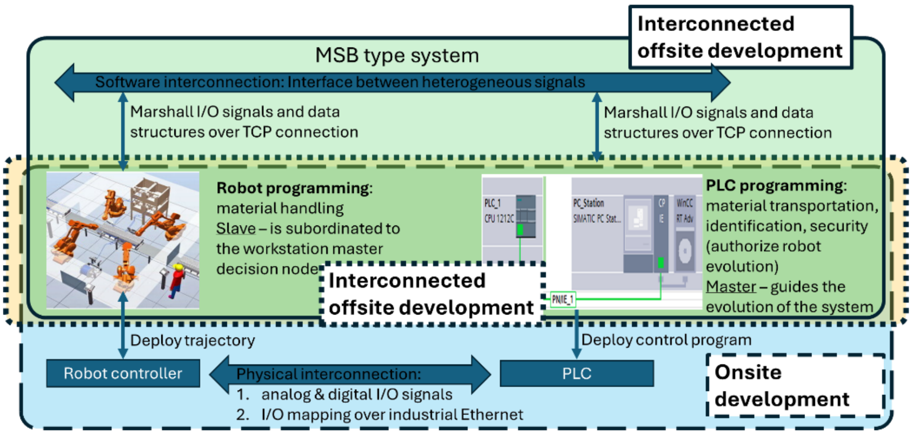

Figure 1. Offsite interconnection architecture (industrial robot DT – PLC DT) – upper section

At the foundation of this architecture lie data streams that facilitate synchronization and information exchange among control units. Each stream is composed of three integral parts: (a) an acquisition module tailored to the functionalities of the originating digital model, (b) a transmission module responsible for writing data to the destination entity, and (c) a centralized software layer that mediates between disparate signal types and a unified, standardized structure. The final component is realized via an ESB-inspired framework, specifically customized for the operational demands of industrial process control systems - referred to as the MSB.

Within the context of a generalized MSB framework, specialized connectors accompanied by protocols are used to structure and encode transmitted information to destination hardware. The selection of approach is governed by the communication capabilities inherent to the system. For example, in PLC-focused development environments, industrial communication protocols enable the mapping of both signals and numeric parameters, while in Digital Twin platforms, programming and simulation are frequently accomplished via raw TCP/UDP protocols operating atop virtual machines with full operating system support.

The protocol delineates a uniform schema for the organization and exchange of internal data within the robotic architecture, thereby ensuring robust and effective integration with external control apparatuses.

This standardized data structure encompasses several key categories:

1. Motion Data: In accordance with established conventions, all information related to robot movement—such as trajectory paths, position points, tool and workpiece reference frames, and load parameters—is centrally stored within the primary program module.
2. Computational and State Variables: Elements such as numeric and Boolean variables used for decision-making processes or temporary storage are standardized to reside within the persistent data module.
3. Maintenance Trajectories: Points or data relevant to maintenance operations are encapsulated solely within the Maintenance module.
4. Routine-Specific Data: Variables or trajectory points relevant exclusively to a particular operational sequence must be declared on a local basis.

Several representative application scenarios can be distinguished, each requiring interconnections established via either conventional hardwired logic (physical I/O) or sophisticated safety-oriented fieldbus protocols. These interconnections must be effectively managed between the corresponding Digital Twins (for example, those of the robot and PLC). Safety Exchange - implementation of safety communication to grant or restrict trajectory execution when potential collisions are detected. Robot-PLC Interaction - facilitating communication between the robot and PLC to specify cycle codes, enabling dynamic selection or parameterization of robot motion paths as required by the production process. Integration is made with Robot-Controlled Peripheral Devices such as end effectors, I/O modules, pneumatic systems, laser sensors or similar devices functioning as subordinate robotic components.

3 Case Study: Experimental scenario, Results and Discussion

A practical case study utilizes the ABB RobotStudio digital twin environment [5] in conjunction with Siemens TIA Portal [6], incorporating the Advanced Simulation module and WinCC for visualization. Conventionally, such automated workstation integrations are performed onsite owing to the inherent physical limitations of connecting control hardware via direct I/O signalling. In this approach, we propose orchestrating synchronization flows through Node-RED, deployed as a Manufacturing Service Bus (MSB)-like middleware. Device connectivity is facilitated by dedicated plugins that leverage the respective system capabilities: RobotStudio interfaces with Node-RED [7] via a TCP protocol, while TIA Portal establishes communication through S7 input/output controls.

Synchronization between the robotic cell and its environment is achieved via logical I/O signals exchanged with the programmable logic controller (PLC). Notably, these signals function as both inputs and outputs depending on the vantage point—for instance, a digital output (DO) from the robot serves as a digital input (DI) for the PLC. For consistency, these are classified as 'orders' (signals issued by the robot to the external environment) and 'events' (signals received by the robot from the environment), with the robot serving as the reference point. Orders and events are programmable at designated positions along the robot's trajectory, either at defined stop locations or during path transitions, using specialized instructions. Illustrative examples of these information flows are presented in Figure 2, contrasting the digital integration with conventional physical I/O-based setups. Figure 3 further demonstrates the connection of a robot digital twin to a PLC simulator.

A control system for a robotized workstation was implemented offsite for experimental purposes. This system utilizes digital signaling to manage the robot's operational flow. The operator selects the required movement trajectory via a Human-Machine Interface (HMI) panel. This selection generates specific digital signals, which are initially transmitted to a programmable logic controller (PLC) functioning as the central decision-making unit. The PLC subsequently forwards these signals to the robot using Node-RED, a flow-based programming tool that enables communication between devices in industrial automation. [7]

Prior to initiating the robot's motion, the PLC conducts a validation check of the robot's readiness status. This process involves confirming that the robot is not currently executing another task and that it is in a safe and prepared state to execute a new trajectory command. Once these conditions are satisfied and a trajectory has been selected by the operator, the PLC authorizes the robot to proceed with the trajectory execution.

Upon activation, the robot performs a predefined pick-and-place operation, following the trajectory selected via the HMI. This integrated control mechanism ensures that commands are executed safely and accurately, and only under appropriate operating conditions. As a result, both the reliability and flexibility of the system are enhanced.

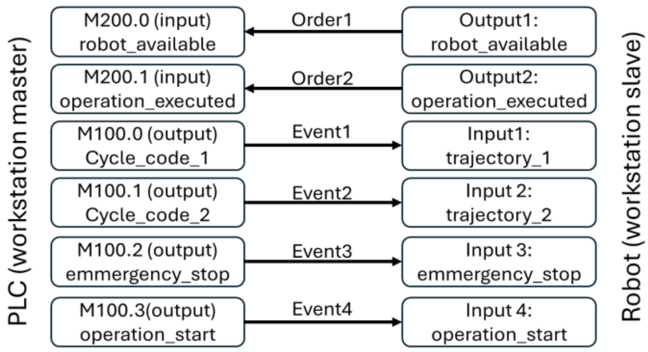

Figure 2. Onsite System Interconnection Architecture

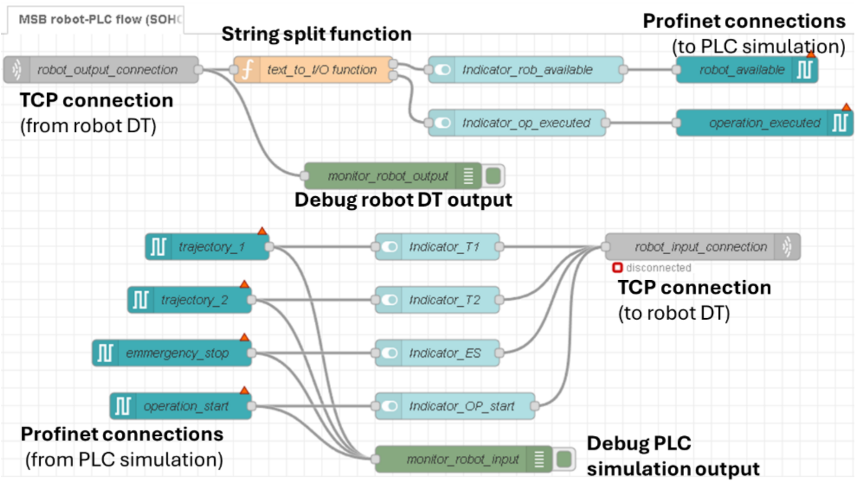

Figure 3. MSB Architecture for Offsite Interconnection

To ensure correct experimental operation, the PLC logic incorporates a series of checks. To determine robot availability, the system verifies the absence of active errors and confirms that the robot is ready for the next task by checking the digital signal M200.0.

Subsequently, to confirm successful completion of the previous task, the PLC checks signal M200.1. This signal is activated by the robot after the previous trajectory has been executed correctly and the robot has returned to its starting position.

When both 'Order 1' and 'Order 2' conditions are TRUE, the operator may select one of the two available trajectories (M100.0 for Trajectory 1 or M100.2 for Trajectory 2) and press the start button (M100.3) to initiate the process.

An additional safety signal is maintained as active (True) throughout the trajectory. In the event of a security breach, such as a human entering the robot's perimeter, the signal is set to False, which triggers the robot Emergency Stop (ES). If the ES button (M100.2) is pressed during operation, the robot immediately halts in its current position with its grippers closed. To resume operation, all hazardous conditions must be eliminated, the physical ES buttons and safety interlocks, triggered by the programmed logic, must be reset, and the robot must be returned to its initial state. After these steps, the system repeats the operational checks described previously.

To monitor the number of parts processed for each trajectory, a counter records the completion frequency of each specific trajectory. A reset button is provided at the bottom of the screen. If the operator does not manually reset the counter, it will automatically reset when the trajectory is changed and will restart from zero.

All interface buttons and controls are linked to the signals defined in Fig. 2 as follows: Available – robot_available (red – false/not available, PLC cannot send commands to robot; green – true/available, PLC can send commands to robot); Operation DONE – operation_executed (red – false/robot on trajectory, PLC cannot send commands to robot; green – true/robot waiting for commands from PLC); Traject i – Cycle_code i ( ); START – operation_start; STOP – emergency_stop. This solution and its implementation enable the development of the PLC program and HMI offsite and facilitate integration with the robot system in simulation. All PLC variables are shared individually via Profinet with the MSB-type system using S7 input/output nodes.

Figure 4 presents a graphical representation of the offsite interconnection between the robot digital twin and the programmable logic controller (PLC) programming and simulation application. This suite of applications, executing the specified pseudocode, enables offsite testing of control programs for workstation upgrades. Following tests of signal synchronization between control applications, sub-second communication times were achieved.

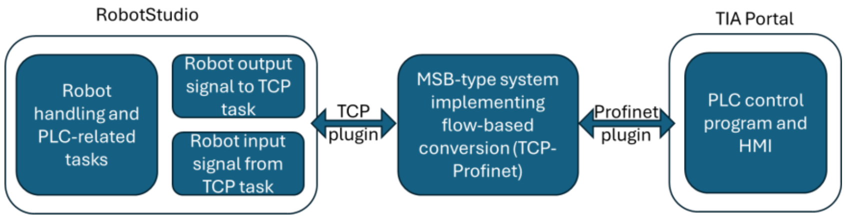

Figure 4. Robot DT - PLC Simulation Environment Offsite Interconnection

4 Conclusions and Future Developments

This study presents a comprehensive solution for the remote integration of control applications used by robotic workstation resources. The research focuses on resources such as industrial robots for material handling and programmable logic controllers (PLCs) for managing product display, identification systems, workstation safety, and operation selection.

The main advantage of the MSB-type system is its versatility, especially in supporting the integration of plugins for data reading and writing and facilitating the development of custom code within defined data flows.

Furthermore, the system allows end-users to interconnect data flows between devices that cannot communicate directly due to functional constraints. For instance, it can connect a machine vision system accessible only via MQTT with a PLC system accessible exclusively through Profinet.

R E F E R E N C E S

[1] Majstorovic, V.D., Mitrovic, R. (2019). Industry 4.0 Programs Worldwide. In: Monostori, L., Majstorovic, V.D., Hu, S.J., Djurdjanovic, D. (eds) Proceedings of the 4th International Conference on the Industry 4.0 Model for Advanced Manufacturing. AMP 2019. Lecture Notes in Mechanical Engineering. Springer, Cham, https://doi.org/10.1007/978-3-030-18180-2_7

[2] Kagermann, H., Wahlster, W., Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group. Acatech – National Academy of Science and Engineering, https://www.din.de/resource/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf

[3] Herlyn, W. (2023). The Digital Twin - a new concept for managing production and logistics in the context of Industry 4.0, WI23 18th International Conference on Wirtschaftsinformatik 18 - 21 September 2023, doi: 10.13140/RG.2.2.21503.79522

[4] Schel, D., Henkel, C., Stock, D., Meyer, O., Rauhoeft, G., Einberger, P., Stöhr, M., Daxer, M., Seidelmann, J. (2018). Manufacturing Service Bus: An Implementation, Procedia CIRP, Vol. 67, pp. 179-184, doi: 10.1016/j.procir.2017.12.196.

[5] Maiolino, P., Woolley, R., Branson, D., Benardos, P., Popov, A., Ratchev, S. (2017). Flexible robot sealant dispensing cell using RGB-D sensor and off-line programming, Robotics and Computer Integrated Manufacturing, Vol. 48, pp. 188-195, doi: 10.1016/j.rcim.2017.04.004

[6] Balla, M., Haffner, O., Kučera, E., Ciganek, J. (2023). Educational Case Studies: Creating a Digital Twin of the Production Line in TIA Portal, Unity, and Game4Automation Framework, Sensors, Vol. 23, no. 4977, doi: 10.3390/s23104977

[7] Ferencz, K., Domokos, J. (2020). Using Node-RED platform in an industrial environment, Jubileumi Kandó Konferencia, file:///C:/Users/ThB/Downloads/UsingNode-REDplatforminanindustrialenvironment.pdf