Special Session: D-MATE: A Design Methodology for Connecting Automatic Test Equipment in Industry 4.0

Special Session: D-MATE: A Design Methodology for Connecting Automatic Test Equipment in Industry 4.0

Francesco Biondani[*], Francesco Tosoni[*], Nicola Dall'Ora[*†], Enrico Fraccaroli[*],
Sara Vinco[‡], Dong Seon Cheng[*], Franco Fummi[*]
[*]Department of Engineering for Innovation Medicine (Section of Engineering and Physics)
– University of Verona (Italy), {francesco.biondani_02, name.surname}@univr.it
[†]Department of Engineering Sciences – Guglielmo Marconi University (Italy), n.dallora@unimarconi.it
[‡]Department of Control and Computer Engineering – Politecnico di Torino (Italy), sara.vinco@polito.it

***Abstract*—With the growing presence of semiconductor devices in healthcare, automotive, and consumer electronics, Automatic Test Equipment (ATE) systems play an increasingly vital role in ensuring quality and reliability during validation. Despite their importance, ATE systems often operate in isolation from other manufacturing processes, limiting interoperability and integration potential. Consequently, fully incorporating ATE systems within the Industry 4.0 framework remains a largely unaddressed challenge. To bridge this gap, we propose adopting Open Platform Communications Unified Architecture (OPC UA), the industry de-facto standard communication protocol for machines, with an accompanying specification tailored to ATE systems. We developed and validated our information model on an advanced ATE system, demonstrating its practical application. The results showcase the successful integration of the ATE system into a fully-fledged Industrial Computer Engineering (ICE) laboratory demonstrator. This study validates the effectiveness of our model in a real-world scenario and highlights the significance of our integration approach within the context of Industry 4.0.**

***Index Terms*—Automatic test equipment, Manufacturing automation, OPC UA, Circuit testing.**

I. Introduction

In the era of rapid technological advancement under Industry 4.0, the demand for reliable semiconductor devices has surged across critical sectors like automotive, avionics, and healthcare. In these safety-critical fields, even minor component failures have serious consequences. As Chris Miller highlights in *Chip War: The Fight for the World's Most Critical Technology* [1], semiconductors are the backbone of modern technology, and ensuring their quality and reliability is paramount. Applying Automatic Test Pattern Generation (ATPG) to these semiconductor-based devices helps identify and intercept defective units. Machines that perform this operation automatically on an industrial scale are called ATE.

The work has been founded by the European Union's - Next Generation EU, Mission 4 Component 1 CUP B31I23000830004. It has been also partially supported by the PRIN 2022T7YSHJ SMART-IC - Next Generation EU project and by the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 101109243. This manuscript reflects only the Authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

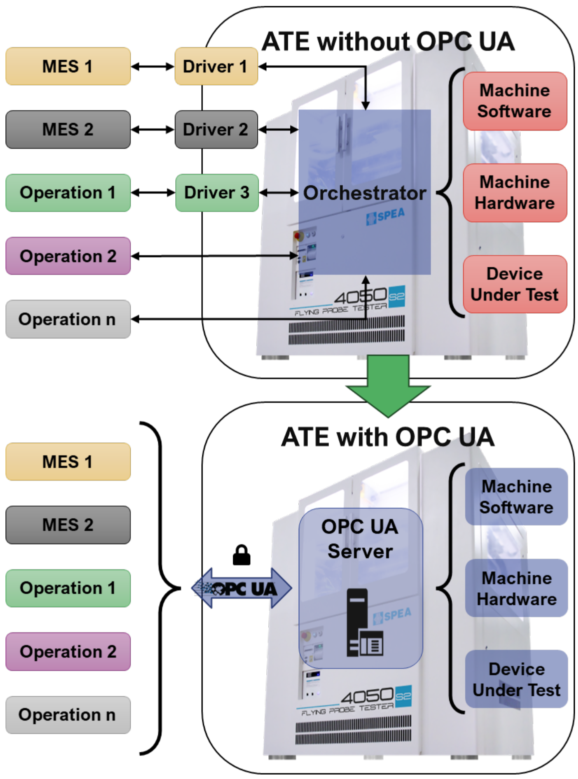

Figure 1. This architecture uses OPC UA as the central interface for accessing machine data and controlling operations. With OPC UA, all external communications—previously non-standardized and customized for different stakeholders—are unified under a single TCP/IP interface, enabling streamlined data transmission improving management and interoperability.

These machines are essential for ensuring the quality and reliability of semiconductor products. ATE machines validate the functionality of chips and electronic components, improving product safety and operational integrity.

Despite their critical importance, ATE machines remain underexplored in the context of Industry 4.0. As chip production scales to meet the demands of interconnected smart technologies, the need for standardized, efficient, and automated testing processes grows. The main challenge is integrating ATE seamlessly into the broader industrial automation architecture, commonly referred to as the *automation pyramid*.

Motivations for this paper: Although Industry 4.0 emphasizes smart manufacturing and interconnected systems, seamlessly integrating ATE machines into the automation pyramid remains challenging. Traditional ATE systems often operate in isolation from other manufacturing processes, resulting in inefficiencies and a lack of interoperability. The lack of standardized communication protocols for ATE, especially in Printed Circuit Board (PCB) testing, limits their potential to meet Industry 4.0 goals like interpretability and information transparency.

Innovations in this paper: This paper addresses the integration gap by introducing D-MATE, a novel approach that adopts the OPC UA standard and proposes a Companion Specification for ATE in PCB testing (see Figure 1). The primary innovations of this work are as follows:

1) Integration of OPC UA with ATE: We present a methodology for incorporating OPC UA into ATE communication architectures, enhancing interoperability and integration within Industry 4.0 ecosystems;
2) Development of a new Companion Specification for ATE: We present a detailed Companion Specification specifically tailored for ATE machines in PCB testing, standardizing information models and communication protocols to improve operational efficiency and consistency.
3) Validation in a real-world scenario: We implement and test the proposed approach in the ICE Laboratory, demonstrating its practicality and effectiveness in an industrial setting.

The remainder of this paper is organized as follows: Section II provides background on OPC UA and Companion Specifications. Section III outlines the methodology for connecting the ATE to a fully interconnected production line. Section IV presents the application of the methodology in a real case study, and Section VI discusses the results. Finally, Section VI provides concluding remarks.

II. OPC UA Background

OPC UA is a flexible, platform-independent, service-oriented architecture compatible with diverse hardware and software environments (see Figure 2). OPC UA effectively integrates industrial devices with cloud systems, microcontrollers, and various Internet of Thing (IoT) applications. Its client-server architecture functions seamlessly across diverse platforms—including Windows, Linux, and mobile operating systems—making OPC UA ideal for the dynamic and rapidly advancing field of industrial automation [2], [3], [4], [5].

The OPC UA framework includes essential components that enable secure, adaptable, and reliable communication. At its core are the *Information Model* [7], *Address Space* [8], and *Services* [9]. These elements standardize object representation and interactions between servers and clients. The *Information Model* offers a scalable structure for defining data types and their relationships, essential for representing complex

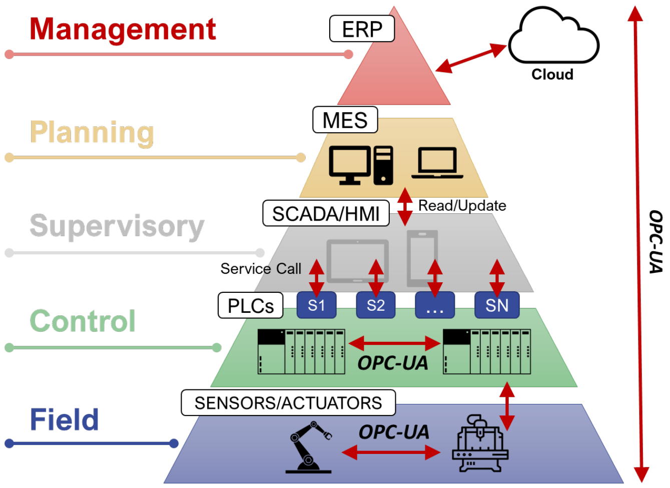

Figure 2. The figure shows the automation pyramid and highlights the critical role of OPC UA across its levels, enabling data communication from field sensors and actuators to ERP systems in the cloud [6].

industrial processes [10]. The *Address Space* organizes data and services so clients can effectively navigate and manipulate them. As a service-oriented protocol, OPC UA relies on its *Services* component to provide fundamental operations that manage client-server interactions. These services include `Discovery`, which enables clients to locate servers (IP address and port), and `SecureChannel`, which establishes secure communication channels. Together, these components support binary and XML encodings, ensuring versatility across various industrial applications.

A. Security in OPC UA

The popularity of OPC UA largely stems from its robust security model (see Figure 3), which incorporates multiple encryption protocols to ensure data confidentiality and integrity. The currently supported **Security Policies** include:

- `None`: this policy applies no encryption or authentication.
- `Basic256Sha256`: uses the SHA-256 hashing algorithm with 256-bit encryption to secure data.
- `Aes256-Sha256-RsaPss`: combines AES-256 encryption with SHA-256 hashing and RSA-PSS for digital signatures.
- `Aes128-Sha256-RsaOaep`: uses AES-128 encryption, SHA-256 hashing, and RSA-OAEP for encryption and authentication.

A key aspect of the security model is **User Authentication**, which supports the following procedures for authentication:

- `AnonymousIdentityToken`: no authentication is required.
- `UserNameIdentityToken`: via username and password.
- `X509IdentityToken`: via X509 certificates.
- `IssuedIdentityToken`: via an external OAuth2 server.

These security mechanisms provide the flexibility and robustness essential for operation in industrial environments, where data protection and access control are paramount.

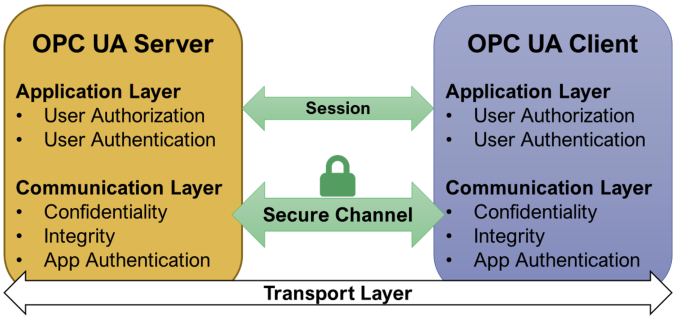

Figure 3. This figure shows the OPC UA security architecture [11], as outlined in OPC 10000-6 [12]. The architecture supports various security objectives through different mappings, enabling security measures at multiple levels.

B. *Companion Specifications*

A core feature of OPC UA is its ability to model complex data and processes in a standardized format using Information Models [13]. These models define data structure and exchange between devices, ensuring interoperability across systems. Industry groups and standardization bodies develop Companion Specifications to provide domain-specific guidelines on applying OPC UA [14]. For example, in the context of ATE, a Companion Specification can define the precise structure and semantics needed to standardize the communication of testing parameters, results, and machine states. Companion Specifications ensure that different ATE machines, even from various manufacturers, communicate using the same standardized language, promoting interoperability and scalability. Adhering to these specifications enables manufacturers to ensure compatibility with a wide range of systems, reducing integration costs and improving overall system reliability.

C. *UA-.NETStandard Library*

The UA-.NETStandard library is an open-source .NET implementation of the OPC UA specification, designed to enable secure, platform-independent communication for industrial applications. We chose the UA-.NETStandard library for its complete support of the OPC UA specification, including transport protocols like TCP and HTTPS. The library supports both binary and XML encodings, making it versatile for developing scalable OPC UA applications. Additionally, the UA-.NETStandard library uniquely enables server timeout configuration through XML files or `ApplicationConfiguration.cs`. The UA-.NETStandard library is recognized as a leading open-source OPC UA implementation for its scalability [15]. It ensures the creation of fully compliant applications that adhere to the latest OPC UA standards. Notably, the UA-.NETStandard library fully supports all the security features specified by the OPC UA Foundation, making it ideal for implementing server architecture for ATE machines [16], [17].

III. Methodology

Our D-MATE framework comprises three key stages, depicted in Figure 4 and outlined in this section.

A. *Information Model Creation*

The first stage of the framework involves creating an information model that serves as the system's foundation. This model defines the structure and semantics needed for effective communication between ATE machines and other components in the production environment. The IM standardizes interactions between the ATE and its surrounding systems by mapping all machine operations, test parameters, and data types. In this stage, we ensure that the information model's compatibility with the OPC UA standard, facilitating interoperability and scalability across industrial scenarios.

B. *Server Implementation*

With the information model established, the next step is implementing the OPC UA server. The server is the central hub for managing and exposing the ATE machine's services and data. In this stage, we integrate the information model into the server infrastructure, enabling it to handle client requests, such as initiating tests or retrieving data. The server also includes custom methods for key functions like file transfers and secure communication, providing a robust platform for real-time machine management.

C. *Test in the ICE Production Line*

The final stage validates the D-MATE framework in a real-world industrial setting, specifically on the ICE Laboratory's production line. Here, we deploy the OPC UA server and a general-purpose client to interact with ATE machines in a fully functional manufacturing environment. The production line, with various manufacturing cells and dynamic workflows, serves as a testbed for assessing the system's efficiency, scalability, and flexibility. By executing customized production recipes and monitoring the ATE performance, we evaluate the framework's robustness and practical applicability for seamless integration in an Industry 4.0 setting.

IV. Implementation

This section exemplifies the implementation of the methodology through the design and development of an information model and OPC UA servers for an ATE machine.

A. *Information Model Creation*

In the first phase, to address the lack of standardization in ATE machine communication and operation when handling PCBs, we developed a tailored information model based on OPC UA. This model provides a structured, consistent approach to managing various aspects of ATE machines, ensuring interoperability and efficiency in line with Industry 4.0 requirements. The information model organizes essential details into key categories detailed in this section. Each category specifies essential *Attributes* and *Methods* (if available) to streamline data handling and communication. The proposed information model standardizes communication, operation, and maintenance management of ATE machines in PCB testing. Adopting this model enables interoperability, efficiency, and reliability, aligning industry practices with the Industry 4.0 principles.

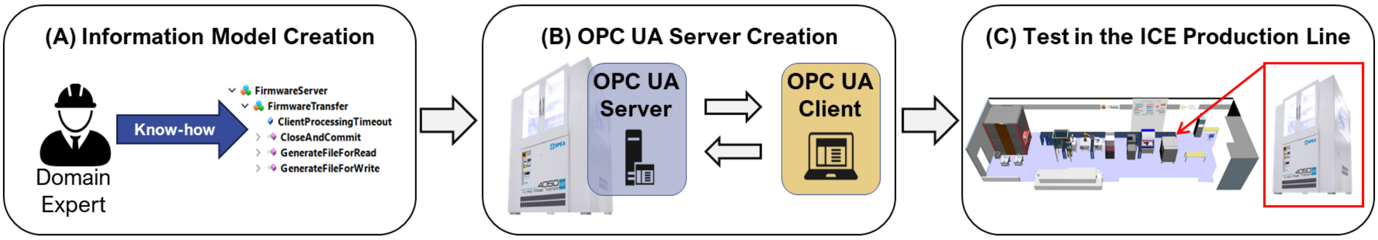

Figure 4. The D-MATE framework begins with the creation of an information model (A), where machine operations and user roles are clearly defined. This is followed by the development of an OPC UA server to expose these operations and a client to simulate a generic user interacting with the system (B). Finally, the server is tested within the Industrial Computer Engineering (ICE) production line using tailored recipes, validating its functionality in a real-world environment (C).

1) ATE Machine Identification: This category provides a unique identifier for each ATE machine and records fundamental details for asset management within the production environment. It contains no methods.

Attributes:

- `MachineID`: A unique identifier for the ATE machine.
- `Manufacturer`: The name of the machine's manufacturer.
- `Model`: The specific model number or name.
- `SerialNumber`: The serial number for traceability.
- `FirmwareVersion`: The installed firmware version.
- `Location`: The machine's physical or logical location within the facility.

2) System Configuration: This component captures the setup and operational parameters of the ATE machine, enabling configuration and connectivity.

Attributes:

- `ConfigurationID`: A unique identifier for the machine's configuration.
- `HardwareSetup`: Details of hardware components and their setup.
- `SoftwareSetup`: Information on software components and their versions.
- `NetworkConfiguration`: Network settings to ensure secure and effective communication.

Methods:

- `LoadConfiguration()`: Loads a specified configuration onto the ATE machine.
- `SaveConfiguration()`: Saves the current configuration for later reference.

3) Test Configuration: This component defines the setup and management of testing processes, enabling operators to establish and control various testing parameters.

Attributes:

- `TestID`: A unique identifier for each test configuration.
- `TestName`: A descriptive name for the test.
- `TestParameters`: Specific parameters and settings for the test.
- `TestSequence`: The sequence of operations required for the test.
- `TestLimits`: Defines acceptable limits and thresholds for test results.

Methods:

- `CreateTestConfiguration()`: Creates a new test configuration.
- `UpdateTestConfiguration()`: Updates an existing test configuration.
- `DeleteTestConfiguration()`: Deletes a specified tes configuration.
- `GetTestConfiguration()`: Retrieves details of a particular test configuration.

4) Test Execution: This component manages each test's lifecycle, enabling real-time monitoring and giving operators control over testing processes.

Attributes:

- `ExecutionID`: A unique identifier for each test execution.
- `StartTime`: The timestamp that indicates when the test starts.
- `EndTime`: The timestamp that indicates when the test concludes.
- `Status`: The current status of the test (e.g., Running, Completed, Failed).

Methods:

- `StartTest()`: Initiates a test based on the specified configuration.
- `StopTest()`: Stops an ongoing test.
- `PauseTest()`: Pauses an active test.
- `ResumeTest()`: Resumes a paused test for operational flexibility.

5) Results Reporting: This component standardizes the recording and management of test results, ensuring consistent documentation for analysis.

Attributes:

- `ResultID`: A unique identifier for each test result.
- `TestID`: References the test configuration used.
- `ExecutionID`: Associates the result with a specific test execution.
- `Timestamp`: The timestamp when the result was generated.
- `ResultData`: Contains detailed data collected from the test.
- `PassFailStatus`: Indicates whether the test passed or failed.

Methods:

- `GetTestResults()`: Retrieves test results based on specified criteria.

- `ExportResults()`: Exports results in standard formats (e.g., CSV, XML).

6) Maintenance and Diagnostics: This category maintains machine health through scheduled maintenance and diagnostics, ensuring reliability and performance.

Attributes:

- `MaintenanceSchedule`: A schedule for regular maintenance tasks.
- `DiagnosticsData`: Data from diagnostic checks on machine performance.
- `ErrorLogs`: Records of errors encountered by the machine.
- `PerformanceMetrics`: Tracks metrics such as uptime and throughput.

Methods:

- `RunDiagnostics()`: Runs a diagnostic check.
- `GetDiagnosticsReport()`: Retrieves a report from diagnostics.
- `ScheduleMaintenance()`: Schedules a maintenance task.
- `GetMaintenanceHistory()`: Accesses the machine's maintenance history.

7) Security Considerations: Given the critical nature of ATE machines, security is a core aspect of the model, ensuring secure communication, authentication, and authorization protocols.

Attributes:

- `UserRoles`: Defines user roles and permissions.
- `AccessLogs`: Logs access and operational activities for monitoring.

Methods:

- `AuthenticateUser()`: Authenticates users based on secure credentials.
- `AuthorizeOperation()`: Verifies if a user is authorized for specific operations.

B. OPC UA Server Creation

After creating our information model, we used the model compiler [18] to convert the XML-based model into source files compatible with our OPC UA server environment. This process ensures compliance with the OPC UA standard and establishes the architecture needed for effective client-server communication. Two primary classes manage the server implementation: `RCServer` and `RCNodeManager`. The `RCServer` class manages overall server properties, node manager creation, and server lifecycle events. Specifically, it configures the server's URI, handles requests, and loads essential properties like manufacturer and software version details. Meanwhile, the `RCNodeManager` is responsible for setting up and maintaining the server's address space. It loads predefined nodes, converts them to the required typed nodes, and links methods to appropriate callbacks, enabling interaction with ATE functionalities.

Our OPC UA server design includes a custom request-handling system to manage prioritized command execution. This handler ensures deterministic management of high-priority commands—particularly those associated with safety-critical operations—by giving them precedence in processing. This system maintains operational integrity, allowing the server to efficiently manage and execute diverse client requests within the production environment.

Figure 5. The SPEA 4050S2 Automatic Flying Probe Tester in the ICE Laboratory. The highlighted area shows a mini-pallet sliding toward the side bay to position a board for testing and programming.

V. Case Study: The ICE Laboratory

The ICE Laboratory at the University of Verona is a state-of-the-art research facility for demonstrating and testing computational technologies for Industry 4.0 and 5.0 applications. The laboratory features a reconfigurable production line with various manufacturing cells, including robotic assembly, quality control, additive and subtractive manufacturing, autonomous logistics, and functional testing.

The facility's functional testing cell incorporates the SPEA 4050S2 Automatic Flying Probe Tester [19], a state-of-the-art ATE system for electronic board testing. This tester is seamlessly integrated into the laboratory's service-oriented manufacturing software architecture and is deployed via a Kubernetes cluster. This architecture framework comprises two core components: the *Data collection architecture* [20] and the *Automation Manager* [21].

A. OPC UA-Based Integration

Each machine in the laboratory, including the SPEA 4050S2, is equipped with an OPC UA server that exposes its services to the *Automation Manager*. The *Data collection architecture* monitors equipment status, stores gathered data, and routes machine commands between laboratory devices and the *Automation Manager* through brokers like RabbitMQ and MQTT. The *Automation Manager* orchestrates production by dynamically executing work orders, responding to real-time changes in the production environment, and controlling the SPEA 4050S2's operations according to predefined recipes.

B. Testing Process

During the demonstration, the SPEA 4050S2 performed various tests on electronic boards, including in-circuit tests, power-on tests, functional tests, boundary scans, and onboard programming. Boards were fed into the machine via a mini-pallet conveyor system, thanks to its seamless integration within the production line. Using the implemented OPC UA server, the ATE was fully integrated into the laboratory's automated workflow, allowing interaction with other cells in the production line (see Figure 5).

The demonstration thoroughly exercised all facets of the companion specification, validating the server's capability to handle diverse operational scenarios. We rigorously tested each model component, from machine identification and configuration to test execution and result reporting, verifying its support for various production needs. Additionally, we evaluated the server's security framework by testing all security levels and access permissions. These tests demonstrated the system's robustness in managing secure communications and controlling access at multiple levels, highlighting the companion specification's flexibility and scalability in an industrial setting.

C. Dynamic Production Scenarios

The laboratory setup demonstrated the flexibility of the ATE in modern production environments by executing different production recipes. In one scenario, the SPEA 4050S2 acted as a testing customer, downloading customized firmware from the board producer's OPC UA server and flashing it onto the electronic boards after successful testing. This setup showcased the production line's ability to accommodate highly customizable workflows, emphasizing the flexibility and efficiency of the OPC UA-based integration.

VI. Discussion and Conclusion

In this paper, we addressed the challenge of integrating ATE machines into the Industry 4.0 framework by adopting the OPC UA standard and developing a dedicated Companion Specification for PCB testing. The proposed architecture was implemented and validated in the ICE Laboratory, demonstrating significant improvements in communication, interoperability, and automation. The integration of OPC UA with ATE in the ICE Laboratory demonstrated significant improvements in data exchange and system interoperability. We standardized communication protocols by proposing a *companion specification*, enabling seamless interactions between ATE and other systems. This interface reduced performance bottlenecks caused by machine isolation, allowing smoother board testing without the need for manual intervention at each test. Unlike traditional ATE systems, which often lack standardized communication frameworks, our OPC UA-based implementation provides a unified architecture, reducing system isolation and the need for manual reconfiguration. This approach represents a significant advancement in enhancing automated testing processes for smart manufacturing environments. Additionally, OPC UA integration lays a foundation for scalable, flexible ATE systems that support real-time data exchange and predictive maintenance.

The Companion Specification serves as a general framework, allowing companies to develop specialized information models tailored to their specific ATE machines and testing environments. This flexibility enables seamless adaptation across different contexts while integrating AI-based analytics, further enhancing automated interpretation of test results and strengthening the system's overall capabilities.

Future work could explore leveraging this system for predictive maintenance on ATE machines. The system could potentially forecast maintenance needs by incorporating real-time data analytics and machine learning models, thereby reducing downtime and enhancing operational efficiency.

Acknowledgment

The authors gratefully acknowledge SPEA S.p.A.'s assistance in acquiring and utilizing the ATE machinery employed in this research. Their contribution facilitated the execution of the research and the development of the presented solution.

References

[1] C. Miller, *Chip war: The fight for the world's most critical technology*. Simon and Schuster, 2022.

[2] J. Luth, "OPC 10000-1 UA Part 1: Overview and Concepts," *OPC Foundation*, 2022.

[3] W. Mahnke *et al.*, *OPC unified architecture*. Springer Science & Business Media, 2009.

[4] O. Givehchi *et al.*, "Interoperability for industrial cyber-physical systems: An approach for legacy systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3370–3378, 2017.

[5] N. C. Găitan *et al.*, "Proposed modbus extension protocol and real-time communication timing requirements for distributed embedded systems," *Technologies*, vol. 12, no. 10, p. 187, 2024.

[6] O. Konradi *et al.*, "Towards an Industrial Converged Network with OPC UA PubSub and TSN," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–4.

[7] J. Harding, "OPC 10000-5 UA Part 5: Information Model," *OPC Foundation*, 2023.

[8] ——, "OPC 10000-3 UA Part 3: Address Space Model," *OPC Foundation*, 2023.

[9] M. Damm, "OPC 10000-4 UA Part 4: Services," *OPC Foundation*, 2023.

[10] S. Cavalieri *et al.*, "Insights into mapping solutions based on opc ua information model applied to the industry 4.0 asset administration shell," *Computers*, vol. 9, no. 2, p. 28, 2020.

[11] P. Hunkar, "OPC 10000-2 UA Part 2: Security," *OPC Foundation*, 2023.

[12] R. Armstrong, "OPC 10000-6 UA Part 6: Mappings," *OPC Foundation*, 2023.

[13] A. Busboom, "Automated generation of opc ua information models—a review and outlook," *Journal of Industrial Information Integration*, p. 100602, 2024.

[14] M. Majumder *et al.*, "Extending the OPC UA Companion Specification for an IEC 61499-based Control System," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2023, pp. 1–4.

[15] N. Mühlbauer *et al.*, "Open-Source OPC UA Security and Scalability," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 262–269.

[16] A. Erba *et al.*, "Security Analysis of Vendor Implementations of the OPC UA Protocol for Industrial Control Systems," in *Proceedings of the 4th Workshop on CPS & IoT Security and Privacy*, ser. CPSIoTSec '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1–13.

[17] F. Biondani *et al.*, "Adopting OPC UA for Efficient and Secure Firmware Transmission in Industry 4.0 Scenarios," in *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*. IEEE, 2024, pp. 1–6.

[18] OPCFoundation, "UA-ModelCompiler," accessed on Feb 27, 2023. [Online]. Available: https://github.com/OPCFoundation/UA-ModelCompiler

[19] SPEA S.p.A., "4050 Automatic Flying Probe Tester," accessed on Feb 27, 2023. [Online]. Available: https://www.spea.com/en/products/4050-s2-flying-probe-tester/

[20] N. Dall'Ora *et al.*, "Digital Transformation of a Production Line: Network Design, Online Data Collection and Energy Monitoring," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 46–59, 2022.

[21] S. Gaiardelli *et al.*, "Enabling Service-Oriented Manufacturing Through Architectures, Models, and Protocols," *IEEE Access*, vol. 12, pp. 85 259–85 274, 2024.