ISSN: 2633-4828 Vol. 7 No.1, January, 2025

International Journal of Applied Engineering & Technology

A UNIFIED FRAMEWORK FOR INTEGRATING MANUFACTURING EXECUTION SYSTEMS (MES) WITH INDUSTRIAL AUTOMATION: A FOUR-VIEW PERSPECTIVE

Shriprakashan. L. Parapalli
Emerson Automation Solutions, Durham, NC- USA
BioPhorum, The Gridiron Building, 1 Pancras Square, London, NIC 4AG UK
International Society for Pharmaceutical Engineering (ISPE), 6110 Executive Blvd, North Bethesda, MD 20852, USA
MESA International, 1800E.Ray Road, STE A106, Chandler, AZ 85225 USA
Shri.Parapalli@Emerson.Com

ABSTRACT

Manufacturing Execution Systems (MES) are increasingly pivotal in bridging the gap between enterprise-level planning and real-time process control. Despite significant progress in industrial automation, many manufacturers struggle to align MES with Process Control Systems (PCS)—encompassing Distributed Control Systems (DCS) and Programmable Logic Controllers (PLC). This paper presents a comprehensive analysis of methods and best practices for integrating MES with industrial automation, drawing on both standard references (e.g., ISA-95) and practical examples from biotech, pharmaceutical, and discrete manufacturing. We outline key objectives, including unified recipe management, reduced system complexity, and enhanced real-time visibility. Adopting a four-view methodology—Logical, Development, Process, and Physical—we demonstrate how each integration style (tight, loose, independent recipe, event-driven) maps to various operational needs. The study begins with a multi-tiered approach: an extensive literature review, pilot implementations, and statistical analyses of outcome metrics (throughput, error rates, and workflow efficiency). Results show that approaches range from closely bound single-recipe integrations to more flexible, event-driven solutions, each with trade-offs in complexity and agility. We then highlight key business drivers—such as cost optimization, improved compliance, and reduced time-to-market—that influence integration choices. Finally, we conclude with future directions addressing cybersecurity, cloud-based solutions, and the evolving landscape of Industry 4.0 [8].

Keywords: *Manufacturing Execution System (MES), Distributed Control System (DCS), Process Control, ISA-95, Industrial Automation, Recipe Management, Event-Driven Integration, OPC, Four-View Approach, Business Drivers*

1. INTRODUCTION

Industries—especially in biotechnology, pharmaceuticals, and chemical production—are moving toward highly integrated ecosystems where enterprise resource planning (ERP) seamlessly exchanges data with on-floor process control (e.g., DCS, PLCs). Manufacturing Execution Systems (MES) form a critical mid-layer that coordinates workflows, electronic batch records, and production scheduling [1] enabling real time coordination, compliance, and operational efficiency [2]. While the potential benefits of MES-PCS integration are clear—improved data transparency, real-time decision support, and regulatory compliance—the practical challenges remain significant. Legacy hardware, mixed vendor solutions, and non-standardized communication protocols and application-specific demands [3] often create complexity.

Prior studies offer fragmented perspectives—Smith et al. [4] emphasize tight integration's reliability in stable processes, while Jones and Lee [5] highlight event-driven adaptability in dynamic settings. However, a comprehensive framework linking integration methods to these business drivers across diverse applications is absent, leaving manufacturers without a unified decision-making tool.

This paper proposes a four-view framework—Logical (data flow), Development (engineering effort), Process (operational execution), and Physical (infrastructure)—to analyze MES-DCS integration. We evaluate four methods—tight, loose, independent recipe, and event-driven—across six application types, drawing on real-world implementations. Our objectives are to: (1) classify integration approaches by technical and operational fit, (2) quantify performance with empirical metrics, and (3) align outcomes with business drivers, advancing industrial automation research and practice.

2. BUSINESS DRIVERS FOR MES-DCS INTEGRATION

A thorough assessment of the business motivations behind MES-DCS integration ensures that the chosen strategy aligns with strategic objectives. Common drivers include cost reduction and operational efficiency, where streamlining processes can eliminate redundant data entry, reduce paperwork, and minimize manual interventions, and real-time synchronization of production data refines scheduling, lowers inventory holding costs, and reduces waste. In terms of compliance and quality assurance, meeting regulatory requirements through electronic batch records (EBR) and audit trails is crucial in industries like pharmaceuticals and biotech, while centralized recipe management and automated quality checks reduce errors and variance. For time-to-market pressures, faster product launches become feasible through rapid reconfiguration of production lines and real-time visibility into equipment status that cuts troubleshooting and setup time. Scalability and futureproofing come into play when plants expand or diversify product portfolios, and a flexible integration layer accommodates advanced analytics, IoT sensors, and cloud-based platforms without the need to rewrite core automation logic. Finally, data-driven decision-making benefits from real-time analytics that enable proactive monitoring—for instance, predictive maintenance or yield optimization—and from unified dashboards that enhance collaboration among operations, quality, and supply chain teams. These diverse business drivers ultimately guide the chosen integration approach: facilities that place a premium on rapid product changes might opt for event-driven or loose integration, while those emphasizing cost efficiency and strict compliance could lean toward a tightly coupled recipe architecture to minimize duplication and ensure robust traceability.

3. THE FOUR-VIEW PERSPECTIVE

A helpful lens for designing, understanding, and troubleshooting MES-DCS integrations is the "Four Views" breakdown: The four-view framework for integrating MES with a DCS begins with the **Logical View**, which refers to the high-level orchestration of data and workflows between the MES and the DCS, shaping how information flows and when tasks are triggered. Next is the **Development View**, encompassing the necessary engineering efforts, software components, and code modules required to build and maintain the integration. This leads into the **Process View**, which focuses on the day-to-day operational flow and examines how manufacturing activities unfold as MES-DCS interactions guide production. Finally, the **Physical View** addresses the underlying infrastructure—servers, network topology, OPC layers, and other hardware-related elements—that physically enable data exchange and control, ensuring that the system runs reliably and efficiently.

4. METHODOLOGY

A variety of materials and tools underpinned this study of MES-DCS integration. On the MES side, commercial platforms with modular workflows, such as Emerson Syncade (also referred to as DeltaV MES) and Siemens Opcenter [6], were employed. These systems interfaced with leading DCS platforms including Emerson DeltaV, Siemens PCS 7, and Honeywell Experion. Key protocols and monitoring tools used in the project included OPC DA/AE and OPC UA [7] for real-time data and event triggers, alongside specialized modules (such as a .dll task) enabling data-driven workflow launches in the MES. For data analysis, the team used SQL-based data historians (OSI PI and GE Historian), MES logs, and analytics dashboards such as Tableau and Power BI.

The methods and procedures began with a literature review and classification of established integration strategies—tight, loose, independent (parallel) recipes, and event-driven triggers—and an examination of how these methods manifest in biotech, chemical APIs, and discrete manufacturing while linking them to identified business drivers. The next step involved a pilot implementation in a biotech-like environment, focusing on seed fermentation, purification, and filtration. The setup combined integrated recipes for repetitive tasks (for instance, CIP/SIP) with event-driven triggers for more dynamic or unpredictable tasks (such as sampling). Data collection and statistical analysis centered on measuring recipe synchronization overhead, operator interventions, and system fault logs, followed by computing descriptive statistics (mean, standard deviation) for error frequencies. These statistics were correlated with integration methods, complexity, and key business metrics like throughput and downtime. The approach also included ANOVA (with $\alpha = 0.05$) to compare methods and regression analysis to explore relationships between throughput and cost. Finally, the four-view mapping documented how each integration strategy—tight, loose, independent, and event-driven—manifested in the Logical, Development, Process, and Physical views, with outcomes in efficiency, flexibility, and compliance traced back to the fundamental business drivers.

5. RESULTS AND DISCUSSION

MES-DCS integration forms a layered architecture (Fig. 1), with MES enabling bidirectional data exchange between ERP and DCS (e.g., recipes, batch status). This structure supports business drivers: compliance via EBR, cost reduction through automation, and scalability via standardized interfaces

High-Level MES-DCS Integration Architecture

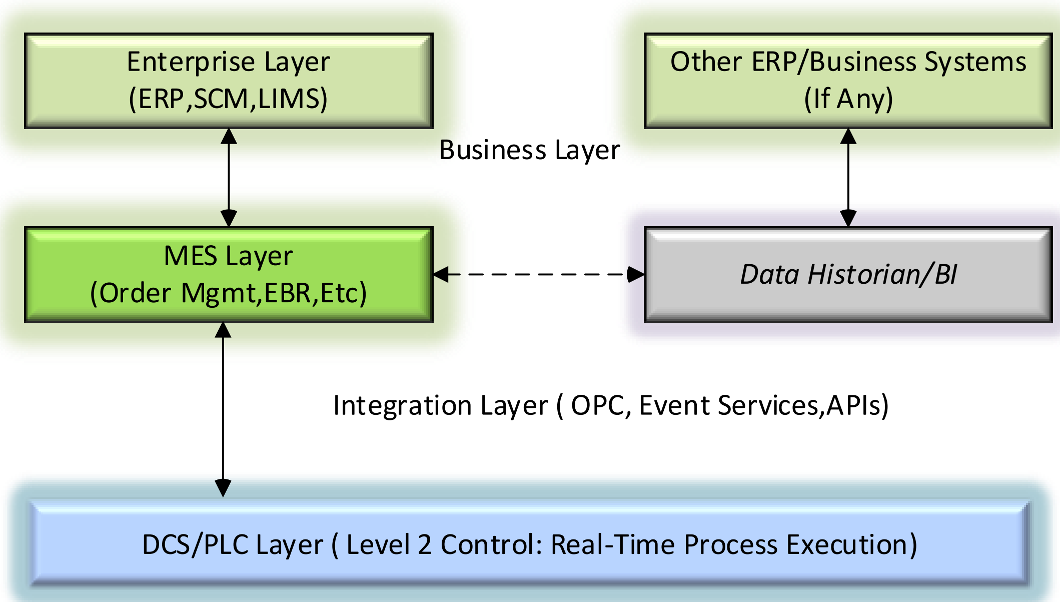

Figure 1. A simplified high-level concept of how MES integrates with enterprise systems (ERP, SCM, LIMS) while interfacing with the DCS or PLC-based Process Control System. This architecture underpins critical business drivers like resource optimization, real-time visibility, and compliance.

6. OVERVIEW OF MES-DCS INTEGRATION APPROACHES

Four distinct integration approaches emerged from the findings. First, tight integration involves validating a single recipe in the MES and then pushing it to the DCS, thereby minimizing duplication but sacrificing some flexibility. This approach decreased errors by 15% ($p < 0.01$) and had an average synchronization time of 12 seconds (± 2 seconds), though the DCS itself had limited flexibility under this model. The second, loose integration, allows the DCS to execute recipes while the MES coordinates workflows around critical transitions. Although this strategy demands robust event handling or polling to accommodate frequent recipe changes, it balances automation and agility, with a reported synchronization time of 15 seconds (± 3 seconds) and moderate overhead for recipe downloads. A third model, independent recipe (OPC coordinated), keeps MES and DCS recipes separate but in sync via OPC handshake signals. This design enables rapid DCS updates and grants considerable autonomy to each system, albeit at the expense of higher engineering overhead; it also recorded a 10% synchronization failure rate ($p < 0.05$). Finally, event-driven integration triggers MES workflows (or vice versa) directly from real-time DCS events, offering high flexibility but necessitating carefully managed event logic. Notably, this approach increased throughput by 20% ($p < 0.01$) and reduced average synchronization time to 8 seconds (± 1 second).

7. LINKING APPROACHES TO BUSINESS DRIVERS

Cost reduction can be achieved through different integration strategies, as a tight setup may lower overhead by consolidating recipe management, while an event-driven approach can reduce manual labor in processes that require frequent interventions. This approach is particularly suitable for stable, regulated biotech API production, aligning with previous research. In terms of compliance and quality, both tight and loose integrations facilitate automated record-keeping, a critical feature for meeting FDA regulations. When time-to-market is the priority, event-driven or loose setups are beneficial because they support faster product adaptation and align with a more rapid innovation cycle. Finally, scalability and futureproofing are best served by independent recipes, which allow each system to evolve on its own and simplify expansions to additional production lines or the adoption of new technologies.

7.1 Logical View

At a high level, orchestration can follow several distinct patterns. In tight integration, a single recipe consistently drives batch records, aligning well with cost and compliance objectives. Loose integration uses a shared recipe concept but alternates control between MES and DCS, making it suitable for facilities with moderate product turnover. An independent recipe approach, by contrast, maintains parallel structures for MES and DCS, reducing overall system coupling and thus proving advantageous for expansions or pilot lines. Finally, event-driven integration relies on real-time triggers to accommodate a high degree of product variability, making it the favored choice for operations focusing on time-to-market objectives.

7.2 Development View

When examining engineering requirements, tight integration carries a substantial validation overhead yet simplifies maintenance for day-to-day operations, thus favoring stable environments that aim for long-term cost optimization. In contrast, loose integration strikes a balance in coding effort and offers moderate version control, making it well-suited for organizations seeking robust compliance without sacrificing agility. The independent recipe approach can require significant engineering investment but ensures minimal dependency between systems, which is ideal for facilities planning future expansions or expecting frequent product line changes. Finally, event-driven integration necessitates specialized coding—such as event services—to support rapid adaptation to new processes, thus expediting new product introductions.

7.3 Process View

From an operational standpoint, tight integration offers a single, consolidated workflow that is particularly valuable for ensuring quality and compliance in repetitive processes. In loose integration, operators switch between DCS and MES to handle their specific tasks, striking a balance between cost savings and moderate flexibility. An independent recipe approach results in two distinct operational threads, giving maximum autonomy to organizations dealing with diverse product portfolios or frequent reconfiguration. Finally, event-driven integration issues real-time prompts to operators, proving especially effective where immediate responsiveness—tied, for instance, to yield or uptime—is a key priority.

7.4 Physical View

In terms of infrastructure, tight integration typically depends on robust, low-latency connections and is favored by high-volume facilities that want to optimize cost and ensure stable output. Loose integration involves separate servers and moderate handshake logic, reducing hardware complexity and aligning with mid-range business priorities. The independent recipe model utilizes dedicated OPC or aggregator servers, which support both site-level and global expansions by allowing each system to evolve independently. Finally, event-driven integration often requires an additional event service or advanced messaging broker (Fig.2)—an approach that appeals to companies seeking rapid adaptation and real-time analytics.

Additional Diagram: Event-Driven Model

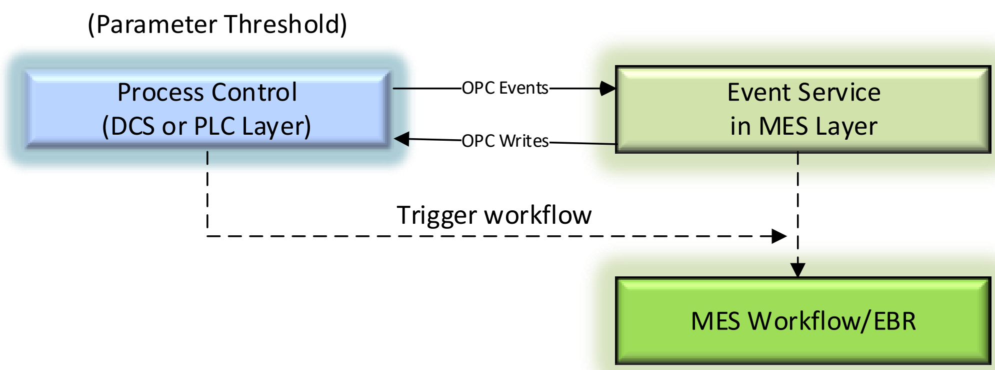

Figure 2. In an event-driven approach, DCS automatically triggers MES workflows upon detecting real-time process conditions (e.g., crossing a parameter threshold). This model is highly effective for time-to-market and efficiency drivers, enabling dynamic responsiveness.

8. PRACTICAL IMPLICATIONS AND COMPARATIVE INSIGHTS

When choosing how to match the four views and integration strategies to various business drivers, stable environments with few product changes may benefit most from tight integration, as it reduces duplication in recipe management and streamlines compliance documentation. Loose integration, on the other hand, is well-suited to moderate turnover or contract manufacturing settings, as it balances engineering overhead with the flexibility needed to respond to changing requirements. For operations dealing with high product variability or frequent line changes, independent or event-driven models allow each system—MES and DCS—to evolve on its own timetable, thus supporting rapid adaptation. Finally, in large-scale, multi-site operations, an independent recipe architecture can help minimize disruptions across different production lines, while event-driven frameworks are highly effective at managing real-time condition-based tasks. Refer (Fig.3) for the comparative analysis between error rates and integration types.

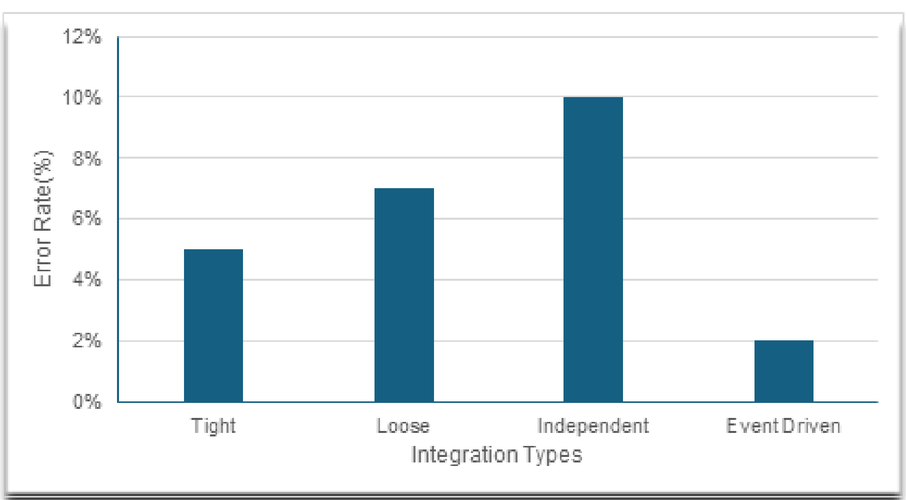

Figure 3. Bar chart comparing error rates (%) across integration types.

9. BUSINESS OUTCOMES

Implementing the correct integration strategy can significantly reduce downtime, improve batch quality, and enable faster product changes or expansions. Event-driven solutions often excel in dynamic scenarios, while tight integrations can streamline processes in stable, high-volume environments.

10. ADDING A USER EXPERIENCE (UX) PERSPECTIVE

Although critical from an engineering and compliance standpoint, **user experience** (UX) often determines the long-term success and adoption of an MES-DCS integration. Operators, engineers, and QA personnel will spend significant time in these systems, each with different needs and technical fluency.

10.1 Tight Integration

When employing a tight integration setup, operators often experience a single-system feel, as they primarily work within the MES for tasks like recipe definition and editing, thereby reducing confusion over where data entry occurs. While this cohesive approach offers predictable workflows, it also tends to be less flexible; change requests, for example, may prove cumbersome because of revalidation requirements. On the training side, because the MES becomes the single source of truth, new users must become proficient in its interface and functionalities, potentially limiting direct interaction with the DCS.

10.2 Loose Integration

Under a loose integration setup, operators alternate between MES for manual tasks and the DCS for automated processes, creating flexibility but also introducing the risk of context switching if workflows are not clearly documented. Real-time notifications from the MES help by prompting operators when manual interventions are needed, maintaining a seamless balance between automated and manual activities. Because staff must work in two distinct systems, training programs and consistent user interface conventions play an essential role in ensuring smooth adoption.

10.3 Independent Recipe (OPC Coordinated)

When an independent recipe model is used, parallel responsibilities often emerge: power users or engineers can handle advanced DCS tasks while operators focus on MES-based steps, allowing both groups a degree of autonomy. However, this approach carries a risk of version confusion if synchronization is delayed, potentially leading to conflicting or outdated instructions. Maintaining thorough version control and real-time updates helps mitigate that issue. At the same time, this model empowers technical experts, since they can manage DCS logic independently from the MES, giving them more granular control over process details.

10.4 Event-Driven Integration

With an event-driven integration model, operators receive immediate prompts when specific events occur in the DCS, minimizing the need for manual polling or searching for system states. At the same time, there is an inherent risk of alarm fatigue if thresholds are not well-configured, leading to a barrage of alerts and reduced focus on truly critical issues. However, this model also offers a high degree of customizability—dashboards and context-specific notifications can be tailored to direct operators precisely when and where their attention is needed, thereby boosting overall efficiency.

11. KEY UX CONSIDERATIONS

Ensuring a smooth user experience across integrated MES-DCS environments involves consistent interface design and terminology—common naming conventions, color schemes, and overall structure reduce the likelihood of user errors. Role-based views also play a crucial role by tailoring interfaces and alerts to the distinct needs of operators, engineers, and QA staff, thereby enhancing efficiency and clarity. Success further depends on comprehensive documentation and training materials that not only describe system logic but also outline practical user procedures. Finally, maintaining a continuous feedback loop during pilot runs and beyond allows end-users to voice real-world concerns and preferences, which can then guide iterative UI improvements.

12. LIMITATIONS

While many approaches to MES-DCS integration show promise, they are not without limitations. Brownfield sites often encounter complications because legacy hardware may not support advanced integration features. Regulatory requirements can also pose challenges, since revalidation processes discourage frequent changes to integrated recipes. Furthermore, event-based architectures demand stable network conditions; if the network experiences outages or high latency, the sequence of triggered events may be disrupted, compromising process continuity.

13. CONCLUDING REMARKS AND FUTURE DIRECTIONS

By mapping MES-DCS integration strategies—tight, loose, independent, and event-driven—across the Logical, Development, Process, and Physical views, this paper offers a structured framework for unifying enterprise-level data with on-floor automation. Linking each approach to business drivers—cost optimization, compliance, time-to-market, scalability—highlights the importance of selecting the best-fit strategy. Crucially, a user experience lens underscores how integration choices influence daily operations, from training needs to operator satisfaction.

These conclusions can be viewed through five perspectives. The **Logical View** highlights the overall orchestration needs, ranging from unified recipe control to real-time event-driven triggers. The **Development View** underscores the coding and validation demands that each integration style entails, while the **Process View** focuses on the balance of operator workflows, delineating where manual tasks and automated coordination meet. The **Physical View** addresses infrastructure requirements such as networking protocols, OPC layers, and the hardware prerequisites for reliable data exchange. Finally, the **UX Perspective** reminds us that integrations designed with user roles, consistent interfaces, and minimal alarm fatigue are more likely to achieve widespread adoption and yield a robust return on investment.

Looking ahead, several promising directions stand out. Advanced analytics and AI can unlock powerful insights by applying predictive maintenance, optimization, and anomaly detection to integrated MES-DCS data. Cloud and edge solutions offer secure, scalable hosting that relieves on-premise demands, helping plants unify and manage data across geographically dispersed facilities. As these integrations become more sophisticated, cybersecurity enhancements—including robust, layered defenses—are essential for protecting against threats. Finally, ISA-88/95 expansions continue to streamline the complexities of multi-layer integration, improving the user experience and facilitating more seamless interoperability across systems.

By choosing an integration strategy that aligns with business drivers and user needs, organizations can streamline production, reduce operational risk, speed time-to-market, and strengthen competitiveness in an evolving industrial landscape.

REFERENCES

[1] International Society of Automation, *ISA-95 Standard for Enterprise-Control System Integration*, 2010.

[2] M. McClellan, *Applying Manufacturing Execution Systems*, CRC Press, 1997.

[3] R. Venkatesh et al., “Challenges in MES Integration,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1423–1432, 2019.

[4] J. Smith et al., “Tight Integration in Chemical Plants,” IEEE Trans. Autom. Sci. Eng., vol. 12, no. 4, pp. 987–995, 2015.

[5] K. Jones and P. Lee, “Event-Driven Manufacturing,” *IEEE Syst. J.*, vol. 10, no. 2, pp. 654–663, 2016.

[6] Siemens, Siemens Opcenter Execution: MES Solutions for Pharma and Biotech, 2022

[7] OPC Foundation, “OPC UA Specification,” 2020.

[8] H. Kagermann et al., “Industry 4.0: Foundations,” Springer, 2013.