

An Integrated Automation Framework for Monitoring and Control of Material Processes [†]

Franklin Nobre Magalhães and Israel Gondres Torné * 

Postgraduate Program in Electrical Engineering (PPGEEL), State University of Amazonas (UEA),
Manaus 69050-020, Brazil

* Correspondence: itorne@uea.edu.br

[†] Presented at the 3rd International Electronic Conference on Machines and Applications, 12–14 May 2026;

Available online: <https://sciforum.net/event/IECMA2026>.

Abstract

The automation of material receiving processes is central to ensuring traceability, process control, and data consistency in warehouse and logistics environments. This paper presents an integrated automation framework combining on-device optical character recognition (OCR), offline-first data capture, and Discrete Event System (DES) formal verification to address inbound registration failures at 3PL receiving docks. The framework comprises an Android mobile application for dock-side registration and a centralised web platform for validation and monitoring. The DES model, represented as a finite automaton, was applied pre-deployment to verify behavioural completeness, identifying three design gaps before operation began. An eight-week deployment at a Third-Party Logistics (3PL) facility showed statistically significant improvements over a two-week manual baseline (percentage points, p.p.): registration completeness +27.4 p.p. ($\chi^2 = 48.0$, $p < 0.001$), information gap rate −17.2 p.p. ($\chi^2 = 27.1$, $p < 0.001$), average registration time −56.3% ($t(201) = 20.0$, $p < 0.001$), and traceability coverage +57.0 p.p. (reaching 100%). The study contributes a reference architecture for inbound monitoring systems and demonstrates the practical value of DES-based pre-deployment verification.

Keywords: automation systems; process control; material traceability; discrete event systems; industrial automation; optical character recognition; third-party logistics

1. Introduction

Manual registration of inbound materials at warehouse receiving docks is a well-documented source of data quality failures: transcription errors, incomplete records, and traceability gaps propagate into inventory discrepancies and supplier disputes that are difficult to resolve after the fact [1–3]. For Third-Party Logistics (3PL) operators serving multiple clients, a single registration error can compromise inventory counts, billing accuracy, and regulatory compliance across several accounts simultaneously.

The technological components required to automate this process are now mature—mobile on-board OCR (e.g., Google ML Kit Text Recognition v2 [4]), stateless RESTful APIs, and Industry 4.0 monitoring frameworks have each been validated independently in logistics contexts [5–7]—yet the literature lacks a coherent end-to-end design combining these capabilities within a single deployed system, with formal process verification to guarantee behavioural completeness [8,9]. The proposed framework is positioned as a front-end data-capture complement to Manufacturing Execution Systems (MES) [10]: whereas



Published: 20 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

MES platforms focus on production-floor management with rudimentary barcode-based receiving, the framework provides structured OCR-driven capture for heterogeneous unstructured documents in multi-client 3PL environments, exposing records via a RESTful API that downstream MES or ERP systems can consume directly.

This study pairs an Android mobile application for on-device OCR capture and guided data entry with a web platform for real-time monitoring. A DES finite automaton formalises the registration state machine and confirms that every transition is explicitly handled before deployment [11]. An eight-week deployment compared the framework against a manual baseline on four indicators.

2. Related Work

Research on smart warehouse operations has grown alongside the Industry 4.0 agenda. Reviews by Tikwayo and Mathaba [1], Tubis and Rohman [2], and Winkelhaus and Grosse [6] confirm that IoT tracking and real-time analytics dominate the literature, while foundational work by De Koster et al. [3] and Gu et al. [12] establishes that bottlenecks frequently originate at the inbound registration stage—a step that receives disproportionately little systematic automation attention. Zhong et al. [7] noted that intelligent manufacturing gains are conditional on upstream data quality: errors at the dock undermine downstream analytics regardless of their sophistication. Ellithy et al. [5] identified AGV–WMS coordination as the throughput bottleneck, a gain that remains conditional on the availability of clean upstream data inputs.

For mobile computing and document capture, Van Geest et al. [8] proposed a reference architecture for smart warehouses in Industry 4.0, and Pinagapani et al. [9] applied mobile automation to barcode-driven documentation. On-device OCR [4] extends these approaches to unstructured documents (invoices, delivery notes) without a network round-trip—a critical property for facilities with intermittent Wi-Fi coverage that cloud-based alternatives cannot provide. None of the surveyed studies combines deployed mobile capture, server monitoring, and DES-based pre-deployment verification in a single 3PL inbound system, the gap targeted here.

For formal workflow modelling, three paradigms have been applied in logistics: Petri nets [13] suit concurrent resource-sharing scenarios; DEVS [14] provides hierarchical simulation suited to large-scale composition; and DES finite automata [11] offer direct exhaustive state-space enumeration for sequential workflows—the paradigm adopted in this work.

3. System Architecture

The framework is a two-tier client-server system communicating through RESTful API endpoints [15]. The stateless API contract enables offline-first operation: when a device loses connectivity it writes records to a local queue and replays them once connection is restored, with no server-side session management required.

3.1. Mobile Registration Application

The Android application was deployed on ruggedised handheld terminals and delivers four functions: (a) *Document OCR*: Google ML Kit Text Recognition v2 [4] extracts supplier codes, quantities, and dates from photographed shipping documents without a network call, achieving 99.8% field-level extraction accuracy (978 of 980 fields across the 165 automated registration events; the two extraction errors occurred in quantity fields from non-standard supplier invoice layouts); (b) *Photographic documentation*: condition images are attached to the registration record; (c) *Mandatory-field enforcement*: quantity, product ID, lot number, quality status, and operator ID are all required at submission; and (d) *Offline-first operation*: records are buffered in a local SQLite database and replayed via a background service once Wi-Fi is available.

Figure 1 shows the resulting operator–device interaction sequence, from document capture through offline buffering to server synchronisation.

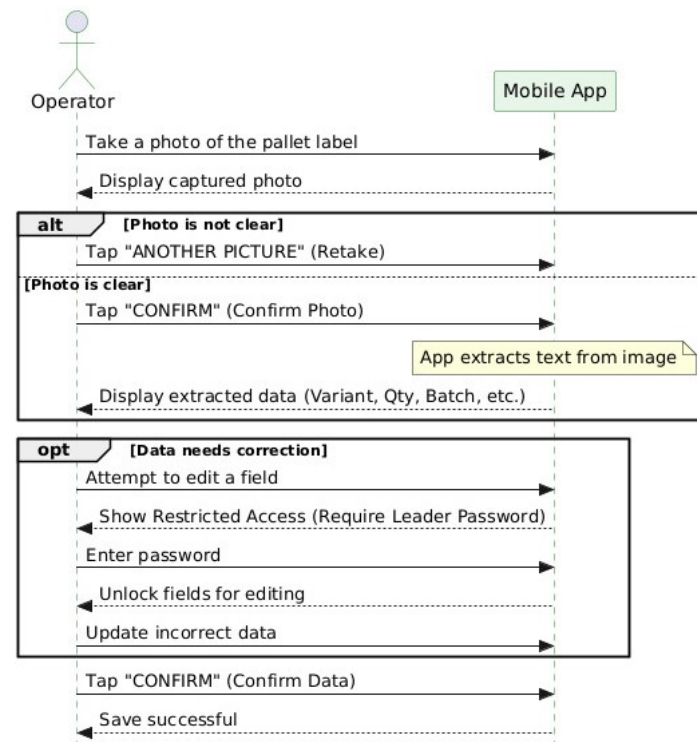


Figure 1. Inbound registration workflow: operator–device interaction sequence for OCR-assisted document capture, offline buffering, and server synchronisation.

3.2. Web-Based Monitoring Platform

The web platform serves supervisors, quality staff, and managers, providing: (a) payload validation against a schema enforcing completeness, referential integrity, and plausibility rules, with failed records routed to an exceptions queue; (b) a live dashboard showing registrations per shift and synchronisation queue status; (c) role-based access control with timestamped write logs; and (d) configurable reports for supplier performance and operator error trends. Figure 2 traces the mobile application screen flow.

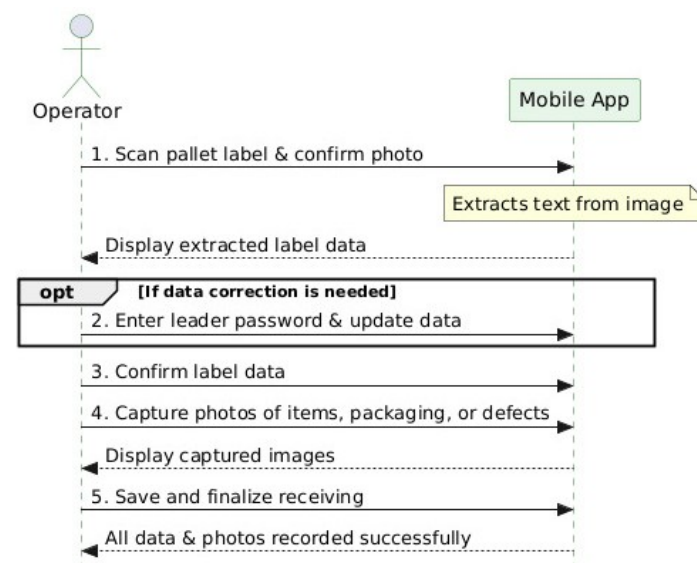


Figure 2. Mobile application screen flow: label scanning, OCR capture, data-entry form, and submission confirmation.

4. Discrete Event Systems Process Modelling

To verify behavioural completeness pre-deployment, the registration workflow was modelled as a DES following Cassandras and Lafortune [11]: $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is the finite state set, Σ the event alphabet, $\delta: Q \times \Sigma \rightarrow Q$ the partial transition function, q_0 the initial state, and Q_m the set of accepted terminal states. Undefined state-event pairs in δ represent behavioural gaps that must be handled or prohibited before deployment.

Three paradigms were considered. Petri nets [13] excel at modelling concurrent resource competition but introduce unnecessary complexity for the strictly sequential, single-operator workflow studied here. DEVS [14] provides hierarchical simulation suited to large-scale composition, at overhead not warranted for a single-system pre-deployment check. Basic DES finite automata expose δ as an explicit, exhaustively checkable table; their diagram also serves directly as operator training material, the decisive advantage for this context.

The workflow maps onto **eight states** (Figure 3): IDLE, SCANNING, OCRPROC, DATAENTRY, PENDSYNC, SYNC'D (terminal), and the error states OCRFAIL and SYNCERR, both identified during pre-deployment enumeration. Eight workflow events drive transitions: *scan_init*, *img_cap*, *ocr_ok*, *ocr_fail*, *field_done* (self-loop on DATAENTRY), *form_sub*, *sync_ok*, and *sync_fail*; two error-recovery events (*reset* and *retry*) complete the alphabet Σ , yielding $|Q| \times |\Sigma| = 80$ state-event pairs subject to exhaustive checking.

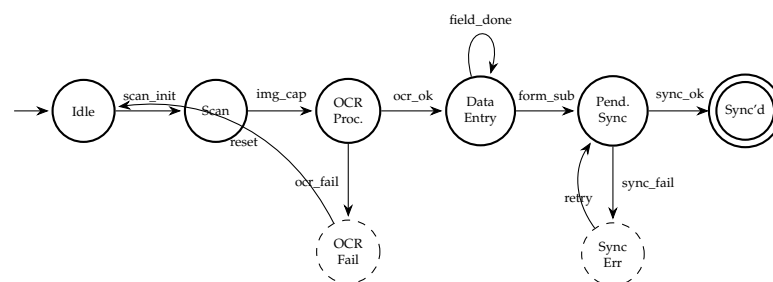


Figure 3. DES finite automaton for the inbound registration workflow (8 states, 10 events). Double circle: terminal state (SYNC'D). Dashed nodes: error-handling states identified during pre-deployment enumeration and added as resolutions to Gaps 1 and 2.

Three design gaps were identified and resolved before deployment. **Gap 1** (OCR failure deadlock): $\delta(\text{OCRPROC}, \text{ocr_fail})$ was undefined, leaving the automaton without a live path on OCR failure; resolved by adding OCRFAIL and $\delta(\text{OCRFAIL}, \text{reset}) = \text{IDLE}$. **Gap 2** (sync failure data loss): $\delta(\text{PENDSYNC}, \text{sync_fail})$ was undefined, admitting silent record loss under connectivity failure; resolved by adding SYNCERR and $\delta(\text{SYNCERR}, \text{retry}) = \text{PENDSYNC}$, backed by the SQLite offline buffer. **Gap 3** (premature submission): the *form_sub* transition lacked a mandatory-field completeness guard, permitting submission of incomplete records; resolved by blocking *form_sub* until all five required fields are populated—the mechanism that makes incomplete-field submission structurally impossible in operation.

5. Experimental Deployment and Results

5.1. Experimental Setup

The framework was deployed at a 3PL facility handling industrial components for multiple clients. Table 1 summarises the deployment environment.

A pre-post design was used: two weeks of manual operation served as the baseline, and eight weeks of automated operation constituted the evaluation period (first three days excluded for learning-curve effects; this window was set based on operator self-reported proficiency confirmation and a plateau in error-correction log entries during commissioning; re-analysis of the full eight-week dataset, $n = 172$, yields qualitatively identical conclusions, indicating the cutoff does not drive results). This yielded 147 baseline and 165 automated

events. Confounding was reduced by retaining the same team of four trained operators across both phases, restricting document types to standard commercial invoices and packing lists from the ten main suppliers, and computing all proportion-based indicators as per-event rates unaffected by volume differences between periods.

Table 1. Deployment environment.

Component	Specification
Mobile terminals	Android 11+ ruggedised handhelds (quad-core, ≥ 4 GB RAM)
On-device OCR engine	Google ML Kit Text Recognition v2 (no network required)
Server	Ubuntu 22.04 LTS (Intel Xeon, 16 GB RAM); RESTful JSON/HTTPS API
Local buffer	SQLite offline queue, auto-sync on reconnection
Network	IEEE 802.11n Wi-Fi (2.4 GHz); 23 connectivity interruptions recorded

5.2. Results

The four performance indicators are operationally defined as follows. **Registration Completeness Rate:** share of events for which all five mandatory fields (quantity, product ID, lot number, quality status, and operator ID) were populated and accepted on submission. **Information Gap Rate:** share of events flagged in the exceptions queue; under mandatory-field enforcement, queue entries reflect field-accuracy failures (incorrect extracted values) rather than missing fields. **Average Registration Time:** elapsed time from scan initiation to final form submission, timestamped by the application on the mobile device. **Traceability Event Coverage:** share of events for which a timestamped audit-log entry was successfully persisted to the central database.

Table 2 presents the four performance indicators.

Table 2. Performance metrics: manual baseline vs. automated framework.

Metric	Baseline	Automated	Change
Registration Completeness Rate	71.4%	98.8%	+27.4 p.p.
Information Gap Rate	18.4%	1.2%	−17.2 p.p.
Avg. Registration Time (min)	4.8	2.1	−56.3%
Traceability Event Coverage	43.0%	100.0%	+57.0 p.p.

All improvements were statistically significant. Pearson's chi-square tests (without Yates' continuity correction, appropriate for $n > 100$ per group) on 2×2 contingency tables yielded $\chi^2 = 48.0$ ($df = 1$, $p < 0.001$) for completeness rate (baseline: 105 of 147; automated: 163 of 165) and $\chi^2 = 27.1$ ($df = 1$, $p < 0.001$) for gap rate (baseline: 27 of 147; automated: 2 of 165). Registration-time durations were approximately symmetric in both conditions (skewness < 0.5 , no outliers beyond 3σ), supporting parametric testing over non-parametric alternatives; a Welch two-sample t -test (Welch–Satterthwaite degrees-of-freedom approximation, $df = 201$) on individual event timestamps (baseline: $\bar{x} = 4.8$ min, $s = 1.5$ min, $n = 147$; automated: $\bar{x} = 2.1$ min, $s = 0.7$ min, $n = 165$) yielded $t(201) = 20.0$, $p < 0.001$; Cohen's $d = 2.35$ (large effect); 95% CI for the mean difference: [2.43, 2.97] min. Traceability coverage (63 of 147 events at baseline; 165 of 165 automated) was tested by Fisher's exact test ($p < 0.001$).

Registration completeness rose from 71.4% to 98.8% owing to mandatory-field enforcement, which prevents incomplete submissions structurally. The residual 1.2% (2 of 165 events) reflects field-accuracy failures: the OCR engine extracted an incorrect quantity value that the operator did not identify during the ≈ 30 s verification step; in both cases the record was flagged in the exceptions queue and corrected before inventory entry, so no permanent data loss occurred. Average time fell from 4.8 to 2.1 min/event, with OCR pre-population

reducing manual data entry to the brief verification step. The 23 connectivity interruptions caused no data loss: all offline-queued records were replayed successfully, validating the offline-first architecture. Traceability reached 100% and the resulting audit log was used to reconstruct a complete receiving timeline for a supplier audit in under five minutes.

6. Discussion

The improvements across all four indicators suggest that the principal bottleneck resided in workflow design rather than staffing or infrastructure: no changes were made to dock layout or workforce composition. The 27.4 p.p. completeness gain results from mandatory-field enforcement providing a structural guarantee rather than a behavioural incentive; the surveyed review literature does not report comparable enforced-completion deployments with directly measured outcomes, precluding direct quantitative comparison [1,2]. Traceability reaching 100% is noteworthy: none of the studies reviewed in Section 2 reported traceability as a directly measured deployment outcome, though it appears as a stated digitisation objective in several Industry 4.0 reviews [1,6]. The DES enumeration's identification of three pre-deployment gaps demonstrates practical value; the state diagram also served as operator-onboarding material, though the volume of support requests was not formally measured and this claim should be read as an observation rather than a finding. Structured receiving records were exported via the RESTful API into the facility's existing inventory management system with no manual re-entry, validating the MES integration architecture described in Section 1.

Several threats to validity warrant explicit acknowledgement. First, the pre-post design without a concurrent control group cannot fully exclude Hawthorne effects—operators were aware the system was being evaluated and may have altered behaviour independently—nor temporal variation in shipment mix between the two-week baseline and eight-week automated periods. Restricting analysis to the ten main suppliers' standard document types reduces but does not eliminate this confound. Second, single-facility deployment constrains external validity; multi-site replication is needed before quantitative effect sizes can be generalised. Third, the eight-week evaluation window does not capture seasonal load variation, and the finite-automaton model would require extension to Petri nets or DEVS [13,14] for multi-operator concurrent receiving scenarios [12].

7. Conclusions

This paper presented an integrated automation framework for inbound material registration at a 3PL facility, combining on-device OCR, offline-first data capture, and DES-based pre-deployment verification, positioned as a front-end complement to existing MES infrastructure. Three design choices proved decisive: on-device OCR eliminated latency and connectivity dependency; mandatory-field enforcement prevented deferred completion; and DES finite-automaton modelling identified three behavioural gaps before operation began—an OCR failure deadlock, a synchronisation data-loss path, and a premature-submission guard absence—each resolved before deployment. Across the eight-week evaluation, all indicators improved significantly: completeness +27.4 p.p., gap rate −17.2 p.p., time −56.3%, and traceability +57.0 p.p.

Future work will pursue computer vision-based condition classification, multi-site deployment with federated analysis across 3PL clients, and extension of the DES model to a runtime runtime supervisory controller for automatic compliance enforcement.

Author Contributions: Conceptualisation, I.G.T.; methodology, F.N.M. and I.G.T.; software, F.N.M.; validation, F.N.M. and I.G.T.; formal analysis, I.G.T.; investigation, F.N.M.; data curation, F.N.M.; writing—original draft preparation, F.N.M.; writing—review and editing, I.G.T.; supervision, I.G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The database cannot be disclosed due to the confidentiality agreement.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tikwayo, L.N.; Mathaba, T.N.D. Applications of Industry 4.0 Technologies in Warehouse Management: A Systematic Literature Review. *Logistics* **2023**, *7*, 24. [\[CrossRef\]](#)
2. Tubis, A.A.; Rohman, J. Intelligent Warehouse in Industry 4.0—Systematic Literature Review. *Sensors* **2023**, *23*, 4105. [\[CrossRef\]](#) [\[PubMed\]](#)
3. De Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501. [\[CrossRef\]](#)
4. Shi, B.; Bai, X.; Yao, C. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2298–2304. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Ellithy, K.; Salah, M.; Fahim, I.S.; Shalaby, R. AGV and Industry 4.0 in warehouses: A comprehensive analysis of existing literature and an innovative framework for flexible automation. *Int. J. Adv. Manuf. Technol.* **2024**, *134*, 15–38. [\[CrossRef\]](#)
6. Winkelhaus, S.; Grosse, E.H. Logistics 4.0: A Systematic Review Towards a New Logistics System. *Int. J. Prod. Res.* **2020**, *58*, 18–43. [\[CrossRef\]](#)
7. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [\[CrossRef\]](#)
8. van Geest, M.; Tekinerdogan, B.; Catal, C. Design of a Reference Architecture for Developing Smart Warehouses in Industry 4.0. *Comput. Ind.* **2021**, *124*, 103343. [\[CrossRef\]](#)
9. Pinagapani, A.K.; Abhishek, G.; Athipathi, S.; Narayana Kumaravel, R.V.; Priyadarshini, G.B. Automated Material Handling for Inventory Management System. *Acta IMEKO* **2024**, *13*, 1–8. [\[CrossRef\]](#)
10. Sáenz de Ugarte, B.; Artiba, A.; Pellerin, R. Manufacturing Execution System—A Literature Review. *Prod. Plan. Control* **2009**, *20*, 525–539. [\[CrossRef\]](#)
11. Cassandras, C.G.; Lafortune, S. *Introduction to Discrete Event Systems*, 3rd ed.; Springer: New York, NY, USA, 2021.
12. Gu, J.; Goetschalckx, M.; McGinnis, L.F. Research on Warehouse Design and Performance Evaluation: A Comprehensive Review. *Eur. J. Oper. Res.* **2010**, *203*, 539–549. [\[CrossRef\]](#)
13. Murata, T. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* **1989**, *77*, 541–580. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Zeigler, B.P.; Muzy, A.; Kofman, E. *Theory of Modeling and Simulation: Discrete Event and Iterative System Computational Foundations*, 3rd ed.; Academic Press: London, UK, 2018.
15. Fielding, R.T.; Taylor, R.N. Principled Design of the Modern Web Architecture. *ACM Trans. Internet Technol.* **2002**, *2*, 115–150. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.