Journal of Guided Motion and Dynamic Energy

ORACLE BASED HOSPITAL INVENTORY SOFTWARE FOR SURGICAL SUPPLIES, PHARMACY STOCK, AND CONSUMABLE TRACKING

Mia Evans
United Kingdom

ABSTRACT

Oracle-based hospital inventory software supports the efficient tracking of surgical supplies, pharmacy stock, and medical consumables through a centralized database platform. The system records item quantities, batch numbers, expiry dates, storage locations, suppliers, purchase details, and departmental usage. Automated alerts notify staff about low stock, near-expiry products, delayed deliveries, and urgent replenishment requirements. Barcode-based tracking can improve the accuracy of receiving, issuing, transferring, and returning inventory items. Integration with pharmacy, procurement, billing, and clinical departments provides real-time visibility into stock movement and consumption. Oracle database technology enables secure storage, rapid retrieval, reliable transactions, and detailed inventory reporting. Overall, the software can prevent shortages, reduce wastage, control procurement costs, and improve the availability of essential hospital supplies.

keywords: Oracle Hospital Inventory; Surgical Supply Tracking; Pharmacy Stock Management; Medical Consumables; Healthcare Inventory Control.

1. Introduction

Enterprise applications depend on reliable diagnostics to detect runtime errors, failed workflows, API issues, slow transactions, and integration faults [1-2]. Logs are the main evidence source for understanding what happened before, during, and after an incident [3]. Effective application diagnostics require structured logging, traceability, contextual metadata, error classification, and operational visibility [4-5]. However, fixed logging strategies often create two problems. Low-detail logs may miss important causes [6], while high-detail logs may generate excessive storage cost and alert noise [7].

The main challenge is that diagnostic needs change during runtime [8]. A normal transaction may require only basic logs, but a failing workflow, repeated exception, or security-sensitive event may need deeper traces [9-10]. Adaptive logging improves this process by changing log depth according to system condition, transaction risk, and error behavior [11-13]. AI-assisted monitoring can identify when additional diagnostic detail is needed and when logging can return to normal levels [14-15]. This article proposes an adaptive logging framework for enterprise application diagnostics.

2. Methodology

The proposed framework captures log metadata from application servers, API gateways, databases, background jobs, authentication modules, and integration services [16]. It records event type, severity, timestamp, user role, transaction ID, request path, error frequency, response time, affected component, and workflow state [17-18]. These features are converted into diagnostic context profiles [19]. Adaptive logging requires context because the same event may be low-risk in one workflow but critical in another [20].

The logging controller classifies runtime states as normal, warning, error-prone, performance-risk, security-sensitive, workflow-critical, or incident-active [21]. Based on this classification, it adjusts log level, trace depth, parameter capture, correlation ID tracking, and diagnostic enrichment. Lightweight scoring and policy rules are used so logging decisions remain fast and explainable [22]. Sensitive fields are masked before storage, and transaction-critical events are linked with workflow and database identifiers for easier diagnosis.

The framework is evaluated using simulated enterprise applications involving APIs, database workflows, authentication events, scheduled jobs, and service integrations. Static logging is compared with adaptive logging under repeated exceptions, slow requests, failed jobs, API timeout, suspicious login, and workflow interruption scenarios [23]. Evaluation metrics include diagnosis time, log volume reduction, root-cause evidence quality, missed-event rate, storage overhead, alert usefulness, and administrator acceptance rate. Feedback from resolved incidents, ignored logs, and administrator notes is used to refine logging policies over time.

3. Results and Discussion

The results show that adaptive logging improves diagnostics by capturing richer evidence only when runtime risk increases. In the baseline condition, static logs either lack enough detail or produce large volumes of low-value records. With the proposed framework, detailed traces are activated for high-risk events and reduced during normal execution. The strongest improvement appears in intermittent failures, where deeper logs are needed only around the failure window.

The discussion shows that adaptive logging must balance diagnostic depth with privacy and storage cost. Excessive logging can expose sensitive data or overload monitoring systems, while weak logging may hide failure causes. The main limitation is that log-level decisions depend on accurate runtime classification. Regular policy review and masking controls are necessary for safe long-term use.

4. Conclusion

This article presented an adaptive logging framework for enterprise application diagnostics. The framework adjusts log detail using runtime context, risk scoring, workflow state, and error behavior. It improves troubleshooting by increasing diagnostic evidence during abnormal conditions while reducing unnecessary log noise during normal operation. The study shows that adaptive logging can strengthen enterprise diagnostics when combined with structured metadata, privacy controls, and continuous incident feedback.

References

1. Keshireddy, S. R. (2019). Spreadsheet-Based Record Migration to Oracle APEX Database Applications. Transactions on Smart Electrical and Computational Systems, 6, 1-5.
2. Gorumutchu, M. R., Jagadhabi, N., Mandapatti, J. K., Bandla, S., & Kavuluri, V. V. R. (2021). Concurrency Anomalies in AI-Mediated Transaction Routing Layers. Journal of Grid, Machines and Drives, 8, 20-26.
3. Keshireddy, S. R. (2022). Secure Session State Management in Oracle APEX Applications. Journal of Privacy-Aware Computing Systems, 9, 8-13.
4. kanth Thottempudi, K., Kande, R., Kotla, C., Nithyanandam, S., & Anjuru, P. (2023). Constitutional AI Governance for Enterprise Analytics: Self-Regulating Architecture for Policy Enforcement, Ethical Constraint Optimization, and Verifiable Compliance. Cross-Disciplinary Knowledge Systems, 10, 29-36.
5. Kavuluri, H. V. R. (2020). Software Reliability Prediction with Weibull Failure Models. Emerging Digital Systems, 7, 7-12.

6. kanth Thottempudi, K., Kande, R., & Kotla, C. (2021). Federated Learning for Privacy-Preserving Clinical Analytics in Multi-Tenant HIPAA Cloud Systems. High-Performance Distributed Computing Review, 8, 28-34.
7. Kavuluri, V. V. R., Gorumutchu, M. R., Bandla, S., Jagadhabi, N., & Mandapatti, J. K. (2024). Query Plan Instability Patterns in Self-Adaptive Optimizers. Transactions on Smart Electrical and Computational Systems, 11, 31-36.
8. Kavuluri, H. V. R. (2023). Version-Controlled Transformation Logic for Reproducible Enterprise Workflows. Journal of Grid, Machines and Drives, 10, 9-16.
9. Mandapatti, J. K., Gorumutchu, M. R., Kavuluri, V. V. R., Jagadhabi, N., & Bandla, S. (2022). Causal Inference Failures in Machine-Learning-Driven Database Tuning. Journal of Privacy-Aware Computing Systems, 9, 1-7.
10. Kotla, C., kanth Thottempudi, K., & Kande, R. (2022). Software Supply Chain Security for Infrastructure-as-Code Pipelines with Vulnerability Detection and Remediation. Perception, Navigation and Intelligent Devices, 9, 29-36.
11. Kavuluri, H. V. R. (2020). Clustering Analysis of User Behavior in Web-Based Software Applications. Cross-Disciplinary Knowledge Systems, 7, 7-12.
12. Keshireddy, S. R. (2023). Anomaly Detection for ETL Execution Graphs Using Temporal Dependency Modeling. Emerging Digital Systems, 10, 9-16.
13. Kavuluri, H. V. R. (2023). Query Pattern Mining for Storage Optimization in Analytical Data Platforms. High-Performance Distributed Computing Review, 10, 9-16.
14. Jagadhabi, N., Mandapatti, J. K., Bandla, S., Gorumutchu, M. R., & Kavuluri, V. V. R. (2022). Semantic Degradation in Long-Lived Machine-Learned Data Pipelines. Transactions on Smart Electrical and Computational Systems, 9, 33-40.
15. Kavuluri, H. V. R. (2020). Oracle Autonomous Database vs Open-Source Solutions: An Overview. Journal of Grid, Machines and Drives, 7, 26-39.
16. Anjuru, P., & Nithyanandam, S. (2024). Event-Driven Architectures for High-Throughput EV Charging Networks. Journal of Privacy-Aware Computing Systems, 11, 30-38.
17. Kavuluri, H. V. R. (2021). Source Code Complexity Assessment with Static Metrics and Maintainability Index. Perception, Navigation and Intelligent Devices, 8, 1-6.
18. Kavuluri, H. V. R., & Keshireddy, S. R. (2019). HIPAA-Compliant Database Security Controls for Cloud-Based Oracle and PostgreSQL Deployments. Cross-Disciplinary Knowledge Systems, 6, 26-33.
19. Mandapatti, J. K., Bandla, S., Kavuluri, V. V. R., Gorumutchu, M. R., & Jagadhabi, N. (2021). Error Propagation Topologies in AI-Assisted Construction Pipelines. Emerging Digital Systems, 8, 39-45.
20. Keshireddy, S. R. (2022). Data Contract Enforcement for Cross-Team Warehouse Pipelines. High-Performance Distributed Computing Review, 9, 1-8.

21. Kavuluri, H. V. R. (2021). Test Case Prioritization with Genetic Algorithms for Regression Testing. Transactions on Smart Electrical and Computational Systems, 8, 1-6.
22. Kande, R., Kotla, C., & kanth Thottempudi, K. (2022). Zero Trust Architecture with MLDriven Behavioral Analytics for Healthcare ERP on Microsoft Azure. Journal of Grid, Machines and Drives, 9, 29-36.
23. Keshireddy, S. R. (2020). Oracle APEX Management Dashboard Development Through SQL Aggregation Views. Journal of Privacy-Aware Computing Systems, 7, 1-6.