ISSN: 2688-8645

A Closed-Loop Alarm Management System for Hazardous Chemical Enterprises Based on Personnel Positioning Data

Haoran Yang

School of East China Jiaotong University, Nanchang, Jiangxi, China

Abstract

Personnel positioning platforms are widely used in hazardous chemical enterprises, but location output alone is still not a safety-control process. In many plant deployments, frequent location snapshots sit apart from rule configuration, alarm archives, response traces, and later review. The scheme uses the output of an existing positioning engine to support closed-loop alarm management. The localization algorithm is left unchanged. The main work is to convert location results into normalized events, electronic geofences, reusable alarm strategies, responsibility-delivery ledgers, response logs, and loop indicators that can be audited. The architecture includes the positioning platform, a business-processing layer, and a front-end management interface. The service layer unifies snapshots, binds tags to personnel objects, performs floor-aware geofence judgment, creates alarm events, records delivery and response evidence, and calculates loop processing time. Anonymized operational samples from alarm, delivery, response, and positioning tables are used for limited verification. The checks cover event-chain completeness, timestamp consistency, calculable closure duration, timeout identification, and ledger consistency. The result is a workflow for turning personnel positioning data into traceable safety-management actions in hazardous chemical plant scenarios, while broader efficiency evaluation requires larger sample disclosure.

Keywords

Personnel Positioning; Electronic Geofencing; Alarm Closed-Loop; Safety Management; Hazardous Chemical Enterprise.

1. Introduction

Hazardous chemical production sites commonly include dense equipment layouts, dangerous media, restricted zones, and complex operating procedures. Risk may increase when personnel enter a prohibited area, remain too long in a high-risk zone, gather beyond the permitted number, or leave a key post unattended. Conventional safety-control methods depend heavily on patrol inspection, manual registration, and post-event record collection. Such methods provide limited support for continuous supervision of moving personnel or for reconstructing the complete evolution of an abnormal event.

Wireless positioning and Industrial Internet of Things technologies make continuous personnel-location acquisition possible, and prior research has described the technical basis of indoor localization and UWB positioning [1, 2]. A positioning engine, however, usually returns coordinates, floor attributes, timestamps, device identifiers, and object states. These are technical outputs, not management actions. If an application only plots points on a map, positioning remains a visualization function. Research on chemical-enterprise safety places more weight on monitoring, process control, risk assessment, and accountability [9-12]. For that reason, location output has to be connected with personnel master data, electronic geofences, alarm rules, responsible objects, delivery ledgers, response evidence, and statistical analysis.

ISSN: 2688-8645

Indoor-positioning studies usually discuss accuracy, signal modeling, filtering, core technologies, and algorithmic robustness [1-4]. Application studies show uses of personnel positioning in chemical production areas, petrochemical plant areas, and safety-management scenarios [5-8]. They say less about what happens after a location snapshot is produced: how it becomes a responsibility-delivery record, an intervention process, and a closed statistical object. In this paper, a closed loop means that a location-triggered alarm has a recorded trigger, receiver, response path, final state, close time, and auditable evidence chain. In many enterprises, the positioning engine is already running. The more practical question is how its output should be organized for daily safety governance. The discussion therefore stays at the application layer, including snapshot normalization, spatial-rule configuration, duplicate-alarm control, and traceable response evidence.

The work has three main parts. It first normalizes positioning events by combining tag identifiers, timestamps, coordinates, floor numbers, object-binding attributes, and online states into one operational event. It then separates electronic geofences from strategy groups, so spatial boundaries and alarm parameters remain independently maintainable and reusable across areas. Finally, it records the alarm process through event records, delivery ledgers, response logs, state-transition constraints, and timeout closure. The evaluation does not test localization accuracy. It checks whether the business records preserve a complete path from positioning output to alarm handling and closure, including timestamp consistency, closure-duration calculation, abnormal-path visibility, and ledger matching. This scope keeps the work separate from localization-algorithm research and from simple map-display applications.

2. System Architecture

Table 1. Core business objects in the proposed system

Object	Main fields	Business role
Positioning event	eventId(PK), tagId, timestamp, x, y, floorNo, status	Standardized input for rule judgment and display
Bound object	tagId(PK), objectType, anonymizedName, bindingStatus	Maps device records to personnel or vehicle objects
Geofence	areaId(PK), polygon, floorNo, validTime, whitelist, version	Specifies areas requiring spatial control
Strategy group	strategyId(PK), alarmType, threshold, receiver, suppressionWindow	Specifies triggering, routing, and suppression logic
Alarm event	alarmId(PK), type, areaId(FK), strategyId(FK), triggerTime, state, closeTime	Primary fact table for alarm statistics
Notification ledger	ledgerId(PK), alarmId(FK), receiver, sendTime, readTime	Stores delivery evidence and receiver status
Handling record	recordId(PK), alarmId(FK), action, operator, handleTime, result	Stores acceptance, transfer, remarks, and closure evidence

The architecture is organized into three layers. The first is the positioning platform, responsible for signal collection, localization calculation, scene management, and location-result output. The second is the business-processing layer, which receives snapshots and conducts object binding, data normalization, rule calculation, alarm governance, ledger writing, and statistical analysis. The third is the front-end interface, supporting personnel monitoring, geofence configuration, alarm response, report query, and three-dimensional situation display.

A central design principle is the separation of localization capability from upper-layer safety-governance logic. The positioning platform continues to produce location results, whereas the application layer determines whether those results constitute a management event. This separation follows the distinction between localization technology and safety-production

ISSN: 2688-8645

monitoring applications discussed in prior studies [1, 10]. It lowers coupling between localization technology and plant-side rules. It also allows the same governance logic to operate with different positioning engines, provided that the required fields are mapped into the unified event model.

At the data layer, stable master data, geofence definitions, strategy groups, alarm events, delivery ledgers, response logs, and audit trails are stored in the business database. High-frequency trajectory points and the latest-position summary may be separated according to query requirements. The summary table is updated by tag identifier and supports rapid display and object search. Historical trajectory records are appended only when storage conditions are met, such as meaningful displacement or later replay requirements. This dual-path design avoids placing every raw point into long-term business tables and supports the data-driven orientation emphasized in chemical risk-analysis research [12].

3. Positioning Data Access and Normalization

Periodic snapshot access is used as the main integration mode. The front end obtains updated personnel positions from an aggregation interface at fixed intervals, while the service layer reads the latest snapshot from the positioning platform. In an enterprise intranet, stability, permission control, and maintainability are more useful than chasing the lowest possible latency. To avoid request accumulation, the next polling request is issued only after the previous one has ended. Timeout control and limited retry reduce the effect of short-term interface fluctuation.

A positioning snapshot usually includes a device or tag identifier, timestamp fields, planar coordinates, floor number, optional longitude and latitude, and online status. A separate binding object gives the tag its operational meaning, such as employee, contractor, visitor, or vehicle. These fields are merged into one positioning event. The event can be displayed on the interface and also used for geofence judgment, alarm-rule calculation, trajectory query, and statistical analysis.

The normalized positioning event can be expressed as a unified tuple:

$$e_i =< u_i, tag_i, t_i, x_i, y_i, f_i, s_i, b_i > \quad (1)$$

where u_i is the business object, tag_i is the device identifier, t_i is the unified timestamp, (x_i,y_i) is the planar coordinate, f_i is the floor number, s_i is the online state, and b_i denotes the binding attribute.

Time normalization is necessary. In the source material, a string-form time field and a millisecond timestamp coexist, and their values may diverge because of time-zone representation. Mixing these fields in retention, stay-duration, or loop-time calculation may produce inconsistent results. The millisecond timestamp is therefore used as the calculation time axis, while the display conversion rule is recorded separately. This handling matches the need for reliable monitoring data in Internet-of-Things safety systems [10]. Formatted time is generated in the user interface, while statistical calculation, event ordering, and state transition use the same temporal basis.

For duration-related rules, the millisecond timestamp is used as the single calculation axis:

$$t_i = \text{timestampMillisecond}_i, \quad t_i < t_{i-1} \Rightarrow \text{mark}(e_i) = \text{abnormal} \quad (2)$$

ISSN: 2688-8645

The service layer also determines whether positioning events are suitable for subsequent calculation. Obvious abnormal records, reverse timestamps, missing identifiers, and mismatched floor information are marked or filtered before they affect geofence rules. The basic validation rules are as follows: each event must contain a tag identifier and timestamp; the calculation timestamp must be later than the previous accepted timestamp of the same tag; the floor number must match the geofence scene; and an unbound tag should not directly generate a personnel-responsibility alarm. The application layer does not replace the positioning engine with a heavy localization algorithm. Instead, it applies lightweight normalization and consistency checks so downstream safety governance receives stable inputs.

4. Geofence-Based Alarm Rule Design

Here, an electronic geofence is treated as more than a geometric boundary. It combines spatial range, floor information, detection period, controlled object type, whitelist, headcount threshold, stay-duration threshold, and strategy-group reference. This model fits personnel-positioning applications in chemical production and petrochemical plant areas, where area control and responsibility assignment are common requirements [5, 6]. The same mechanism expresses restricted-area entry, excessive stay, over-occupancy, missing key post, and static abnormality through different rule types attached to geofences and strategy groups.

The judgment process first screens candidate geofences by scene and floor. In multi-floor or three-dimensional plant environments, identical planar coordinates may correspond to different physical locations. Floor-aware filtering is therefore completed before planar geometry calculation. A bounding-box precheck is then used to avoid unnecessary polygon computation. Accurate point-in-polygon judgment is performed only when a point falls within the bounding box. Boundary tolerance is included so that edge points are processed consistently.

If every point is used directly, short-term positioning fluctuation may lead to repeated or false alarms, a problem also reflected in robustness-oriented positioning research [3, 4]. To mitigate this risk, spatial judgment is combined with time thresholds, displacement thresholds, and suppression windows. For a stay alarm, the first entry time of an object in the monitored area is recorded, and the alarm is triggered only after the configured duration is exceeded. For a static alarm, cumulative movement and elapsed time are evaluated together. For repeated events, an idempotent key consisting of object identifier, geofence identifier, alarm type, and active time window determines whether a new trigger belongs to an existing alarm instance.

The floor-aware geofence membership function is defined as follows:

$$I(e\_i, g\_j) = 1\{f\_i = f\_j\} * 1\{(x\_i, y\_i) \text{ in } P\_j\} * 1\{t\_i \text{ in } T\_j\} \quad (3)$$

A continuous alarm is generated only when spatial membership and temporal persistence are both satisfied:

$$A\_\{i,j\} = 1 \text{ if } I(e\_i, g\_j) = 1 \text{ and } (D\_\{i,j\} >= \text{tau\_j or } N\_j >= H\_j), \text{otherwise } 0 \quad (4)$$

where P_j is the polygon of fence g_j, T_j is the detection period, D_{i,j} is the dwell duration, tau_j is the dwell threshold, N_j is the current headcount, and H_j is the headcount threshold.

Geofences and strategy groups are separated. A geofence specifies where control is required, whereas a strategy group specifies how control is executed once a condition is met. The strategy group stores alarm type, threshold parameters, receiver mapping, suppression window, and

ISSN: 2688-8645

automatic-closure conditions. Multiple geofences may reference the same strategy group. This arrangement reduces duplicate configuration and simplifies later adjustment. When a threshold or receiver changes, similar geofences do not need to be edited individually.

Geofence and strategy settings directly affect online alarm calculation, so configuration changes need their own workflow. The system distinguishes draft, published, and rollback states. Draft changes do not enter online judgment immediately, while published versions are used by the alarm engine. Rollback records and audit logs are retained for later review. This control matters in high-risk areas because accidental changes may cause missed alarms or unnecessary alarm flooding.

5. Closed-Loop Alarm Handling Mechanism

Alarm display is only the starting point. Each alarm has to become a traceable management event from triggering to final closure, which is consistent with the accountability requirements discussed in chemical safety studies [7, 9, 11]. The process is organized by a state machine with six main states: NEW, NOTIFIED, ACCEPTED, PROCESSING, CLOSED, and TIMEOUT_CLOSED. NEW means that a rule condition has been met and an alarm event has been created. NOTIFIED means that the responsible object has been written into the delivery ledger. ACCEPTED and PROCESSING correspond to manual intervention. CLOSED denotes normal closure, while TIMEOUT_CLOSED denotes automatic closure after an overdue threshold. A loop is complete only when the alarm has a final state, close time, and auditable evidence chain. Direct NEW-to-CLOSED closure is treated as abnormal unless a rule records the reason.

Table 2. Alarm state transition model

State	Trigger condition	Recorded evidence
NEW	Rule condition is satisfied	Alarm event
NOTIFIED	Receiver delivery record is generated	Notification ledger
ACCEPTED	Responsible user acknowledges the alarm	Handling record
PROCESSING	Response action is submitted	Response record and remarks
CLOSED	Manual closure is approved	Closure time and result
TIMEOUT_CLOSED	Scheduled task closes overdue event	System response record

When an alarm is generated, the application writes the alarm event and delivery ledger in the same transaction, ensuring that each event has a corresponding delivery record. Subsequent actions, including acceptance, transfer, remarks, false-alarm feedback, and closure, are stored as response records associated with the alarm identifier. The alarm master table stores the current state, close time, and closure reason, while the response table preserves the detailed process. The basic audit rules are: each alarmId should have a delivery ledger; each closed alarm should contain closeTime and at least one response record; notificationTime should not precede triggerTime; and closeTime should not precede acceptTime. This structure supports both real-time workbench display and later statistical analysis.

To suppress duplicate records across polling cycles, the alarm instance is constrained by an idempotent key:

$$k = \text{hash}(\text{tenant}, \text{tag\_i}, \text{g\_j}, \text{alarmType}, \text{floor}(\text{t\_i}/\text{Delta})) \quad (5)$$

ISSN: 2688-8645

A delivery ledger is used because many enterprise intranets do not have instant-communication infrastructure for every safety workflow. The ledger records the intended receiver, generation time, read status, and later state changes. It is less immediate than a dedicated message channel, but it is stable, auditable, and compatible with role-based permission control. External channels can still be added for high-priority events without changing the basic evidence model.

Timeout closure prevents alarms from remaining open indefinitely. A scheduled task scans active events and conditionally closes those exceeding the configured threshold. The task does not silently remove the event; it writes a system response record and changes the state to TIMEOUT_CLOSED. This distinction separates manual response behavior from overdue accumulation. In later analysis, manual closures, false-alarm closures, and timeout closures should be interpreted separately because they represent different management phenomena and support different safety-improvement decisions [11, 12].

6. Application Verification

Verification uses anonymized operational samples drawn from alarm events, delivery ledgers, response records, and positioning records. Because the disclosed material does not include a complete ground-truth measurement set, the paper does not evaluate positioning accuracy. The verification is limited to management-chain evidence: whether the alarm path is complete, whether loop time is calculable, whether time fields can be unified, and whether related tables remain consistent. This matches safety-production management systems that rely on monitoring records, traceability, and risk analysis rather than localization algorithms alone [10-12].

Table 3. Verification items and anonymized findings

Verification item	Method	Finding
Event-chain completeness	Match alarmId across event, ledger, and response tables	Main chain can be matched in disclosed samples
Manual response duration	Calculate closeTime minus triggerTime in minutes	Duration is calculable; full statistics require larger sample disclosure
Timeout closure	Check TIMEOUT_CLOSED records and system handling logs	Timeout path is identifiable through system response records
Time consistency	Compare string time, millisecond timestamp, and display conversion	Timestamp offset is controlled by using millisecond time
Ledger-response consistency	Check ledger existence, closeTime, response record, and state order	Rule violations can be converted into inspection items

The disclosed samples include manual closure, false-alarm closure, and automatic timeout closure. A sample set containing only successful manual intervention would be too optimistic and would leave the fallback path untested. For each case, the check records alarm type, closure path, trigger time, notification time, acceptance time, close time, close reason, and the existence of corresponding delivery and response evidence. Putting different closure paths in the same verification group makes the verification stricter, because the same state-transition and ledger rules must hold for normal handling, false-alarm feedback, and overdue convergence.

Loop duration is measured from the first alarm trigger time to final closure time and then converted into minutes. The disclosed samples are sufficient for checking calculability and table consistency, but not for making strong statistical claims about response efficiency. A full deployment report should disclose the number of alarm events, manual closures, false-alarm closures, and timeout closures, together with descriptive statistics such as minimum, median,

ISSN: 2688-8645

mean, maximum, and timeout count. Manual response duration and timeout duration need separate analysis because they measure different operational phenomena.

The closed-loop duration used in verification is calculated in minutes:

$$L = (t\_close - t\_trigger)/60000 \quad (6)$$

The verification also shows why string-form time and millisecond timestamps should not be mixed. A visible offset may appear when the two fields are compared directly. The millisecond timestamp is therefore retained as the unified calculation basis, while formatted display time is generated separately. This rule affects stay alarms, static alarms, polling freshness, and loop-time statistics. Without one time axis, a record may look correct in the interface but still produce inconsistent analytical results, weakening later safety-risk analysis [12].

Table consistency is checked as well. When an alarm master record exists, a corresponding delivery ledger is expected. Once an alarm is closed, a response record must be traceable. Receiver, permission scope, trigger time, notification time, acceptance time, and close time need to follow a reasonable order. These checks are simple enough to become routine database inspection tasks. They also move safety-management review away from interface observation alone and toward record-based evidence [8, 9].

The verification indicates that positioning output can be organized as a traceable safety-management process under the disclosed sample conditions. It does not claim that one positioning engine is more accurate than another, nor does it prove general response-efficiency improvement. The test is narrower: whether output from an existing engine forms geofence rules, alarm events, responsibility delivery, response evidence, auditable state transitions, and measurable loop statistics. This is the type of record evidence many plant-side safety scenarios need before larger-scale performance evaluation.

7. Discussion

The design is most suitable when a positioning engine already exists and the harder problem is application integration. Its main value is the connection between location data and management evidence. Compared with simple map display, the approach preserves records for alarm generation, notification, acceptance, response, closure, and later review. These records give threshold adjustment and process optimization a clearer data basis and move personnel-positioning applications beyond visualization [7, 8].

Several limitations remain. Verification relies on anonymized operational samples, not a large controlled experiment, so the current evidence supports traceability and calculability rather than broad claims about operational efficiency. The paper also does not compare positioning accuracy, because the work is not an algorithm study. In addition, the polling architecture introduces latency. Future work should track interface response time, data freshness, polling jitter, front-end rendering time, failure rate, false-alarm rate, timeout-closure ratio, and closure-duration distribution over a longer period. These indicators would help separate localization error, interface congestion, network fluctuation, rendering pressure, and management-response delay.

The design may also be used in warehousing, energy facilities, and equipment manufacturing sites after area types and event definitions are remapped. Geofence modeling, strategy configuration, alarm state transition, ledger writing, response evidence, and three-dimensional display are relatively transferable. Risk categories, threshold values, responsibility mapping, and closure rules still need to be defined for each scenario.

ISSN: 2688-8645

8. Conclusion

The paper presents a closed-loop alarm management system for hazardous chemical enterprises using personnel positioning data. The underlying positioning algorithm remains unchanged. The work turns location output into operational evidence, including normalized events, electronic geofences, reusable strategy groups, alarm-state management, delivery-ledger writing, response-record retention, timeout closure, and three-dimensional situation display. Verification with anonymized samples shows that the disclosed alarm chains are traceable from triggering to final closure, loop time is calculable, state-transition evidence is auditable, and time-field consistency is controlled through a unified timestamp. The system gives hazardous chemical enterprises a concrete way to use positioning data in auditable safety-management work, while larger samples are still needed to evaluate response efficiency.

Acknowledgments

The author thanks the academic supervisors and project collaborators for guidance on system design, engineering implementation, and manuscript preparation. All enterprise-related data used in this paper have been anonymized.

References

[1] Zafari F, Gkelias A, Leung K K. A Survey of Indoor Localization Systems and Technologies[J]. IEEE Communications Surveys & Tutorials, 2019, 21(3): 2568-2599.

[2] Alarifi A, Al-Salman A, Alsaleh M, et al. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances[J]. Sensors, 2016, 16(5): 707.

[3] Liu J, Pu J, Sun L, He Z. An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots[J]. Sensors, 2019, 19(4): 950.

[4] Deng Z L, Yin L, Tang S H, et al. Review of Key Technologies for Indoor Positioning[J]. Navigation Positioning and Timing, 2018, 5(3): 14-23. (In Chinese)

[5] Yuan S. Design and Implementation of Personnel Positioning Method and Application Scheme for Chemical Production Areas[D]. Zhejiang University, 2018. (In Chinese)

[6] Wu J. Personnel Positioning Technology and Its Application in Petrochemical Plant Areas[J]. Industrial Innovation Research, 2021, (24): 51-53. (In Chinese)

[7] Yu W L. Application of Personnel Positioning System in Chemical Enterprise Safety Management[J]. Chemical Fiber and Textile Technology, 2024, 53(04): 112-114. (In Chinese)

[8] Zhang J H, Zhu C G, Zhao M. Application of Personnel Positioning System in Safety Management[J]. Shandong Coal Science and Technology, 2009, (03): 85-86. (In Chinese)

[9] Chen C M. Analysis of the Importance of Chemical Safety Management[J]. Chemical Enterprise Management, 2019, (10): 136. (In Chinese)

[10] Wang F, Chen G, Zhao J, et al. Research on Chemical Enterprise Safety Production Monitoring System Based on Internet of Things[J]. Internet of Things Technologies, 2019, 9(11): 45-47. (In Chinese)

[11] Zhang X M, Liu J, Li Q, et al. Design and Implementation of a Chemical Safety Production Management System[J]. Journal of Software, 2020, 31(7): 2023-2038. (In Chinese)

[12] Li H, Wang H, Zhang W, et al. Application of Big Data Analysis in Chemical Safety Production Risk Assessment[J]. CIESC Journal, 2021, 72(12): 6856-6865. (In Chinese)