DECISION RULES FOR LINK, MERGE, AND UPDATE OPERATORS IN AN INCREMENTAL MULTI-AGENT FACT REPOSITORY

D.E. Khujamberdiyev

[1]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan

d.xujamberdiyev@tuit.uz

Abstract. Records arriving from distributed agents may be duplicates, supporting evidence, later states of an existing event, or genuinely new facts. Treating all similar records as duplicates causes information loss, while storing every record separately produces an inflated and poorly auditable repository. This paper formulates decision rules for distinguishing the Link, Merge, and Update operations within an incremental fact-repository method based on five lifecycle operators: Insert, Link, Merge, Update, and Close. The decision is made from referent identity, semantic similarity, temporal proximity, the role of the incoming evidence, and the current lifecycle status. A three-stage procedure first identifies candidate facts, then classifies the relation between the incoming fact and the candidate, and finally records the resulting transition in the repository history. A prototype demonstration on 20,000 service-request records reduced the active representation to 6,319 facts while retaining 19,544 evidential or semantic relations. The results show that repository compactness and evidence traceability can be achieved simultaneously only when Merge, Link, and Update are treated as different operations rather than a single deduplication step.

Keywords: multi-agent system, fact repository, operator selection, semantic relation, duplicate consolidation, evidence traceability, lifecycle state.

INTRODUCTION

In electronic service portals, monitoring systems, machine logs, and video-event streams, several records may refer to the same real-world object without having the same informational role. One record may repeat an earlier complaint, another may add a photograph or a diagnostic message, and a third may describe the completion of the incident. A repository that relies only on append operations cannot separate these cases. The result is a large set of isolated records in which duplicates, supporting evidence, follow-up states, and closed events are mixed together.

Multi-agent systems distribute data acquisition and processing among autonomous agents [1–4]. This architecture improves scalability, but it also makes repository maintenance dependent on local decisions. The agent must determine whether an incoming normalised fact should become an independent fact, be linked to an existing fact, be merged with it, enrich it, or close its lifecycle. The most difficult distinction is

between Link, Merge, and Update. These operations may be triggered by similar records, yet they have different consequences for fact identity, evidence preservation, and subsequent retrieval.

This work formalises that distinction for the third scientific result of the dissertation: incremental repository maintenance through Insert, Link, Merge, Update, and Close. The analysis focuses on the semantic boundary between the three central operations and on their effect on compactness and auditability.

FACT STATE AND THE OPERATOR-SELECTION PROBLEM

A normalised fact is represented by the seven-component structure

$$f = \langle k, s, r, h, t, d, w \rangle, \tag{1}$$

where k is the key, s is the source agent, r is the referent, h is the event semantics, t is the time point or interval, d is the evidence description, and w is the reliability value. At processing step n, the local repository of agent a_i is described by

$$\Lambda_i^n = \langle F_i^n, L_i^n, U_i^n \rangle, \tag{2}$$

where $F_i^{(n)}$ is the fact set, $L_i^{(n)}$ is the set of semantic or evidential relations, and $U_i^{(n)}$ is the ordered transition history. An incoming normalised fact f* is first compared with active facts having the same or a close referent. Insert is selected only when no semantically close active equivalent with the same referent exists:

$$\neg\exists\, f' \in F_i^n : \sigma(f*, f') \geq \theta_m \wedge r(f*) = r(f'). \tag{3}$$

The three ambiguous operations are separated by the informational role of the incoming record. Merge is appropriate when two facts describe the same event unit. It requires the same referent, sufficiently high semantic similarity, and acceptable temporal proximity:

$$r(f_p) = r(f_q) \wedge \sigma(f_p, f_q) \geq \theta_m \wedge \Delta t(f_p, f_q) \leq \theta_t. \tag{4}$$

Link is used when two facts are related but must remain distinct semantic units. In that case the repository keeps both keys and adds a typed relation ℓ, such as continuation, evidence, repetition, state change, or another representation of the same broader incident:

$$L_i^{n+1} = L_i^n \cup \{(f_p, f_q, \ell)\}. \tag{5}$$

Update differs from both operations because it does not create a second semantic unit and does not replace two facts with one. It enriches an existing fact with a new attribute, timestamp, state, or evidence item while retaining its semantic core:

$$f' = f \oplus \Delta, \quad d(f) \subseteq d(f'). \tag{6}$$

Close is applied when the event reaches a terminal state. It changes lifecycle status but preserves the transition history. Consequently, the choice among the five operators is not based only on similarity: it depends on whether the incoming information changes identity, relation, content, or lifecycle state.

OPERATOR DECISION PROCEDURE

The decision procedure is organised as three consecutive checks. The identity check determines whether a compatible active fact exists. The relation check distinguishes equivalence from semantic or evidential association. The lifecycle check determines whether the incoming record enriches the current fact or moves it to a terminal state. This organisation prevents the common error of applying Merge to every similar record.

Table 1. Decision conditions for repository operators

Operator	Dominant condition	Effect on repository
Insert	No compatible active fact exists.	Creates a new fact and key.
Link	Facts are related, but represent distinct semantic units.	Keeps both facts and adds relation ℓ.
Merge	Same referent, high semantic similarity, and close time.	Replaces duplicates by one consolidated fact.
Update	The incoming record adds a state, attribute, time, or evidence item.	Enriches the existing fact without changing identity.
Close	The event has reached a terminal status.	Marks the lifecycle as completed and preserves history.

For each new fact f*, the agent performs the following sequence: (1) select facts with the same or a close referent; (2) estimate semantic similarity and temporal proximity; (3) use Insert when no active equivalent exists; (4) use Link when the relation is meaningful but fact identity must remain separate; (5) use Merge when the identity, semantic, and temporal conditions jointly indicate the same event; (6) use Update when f* extends the current fact; (7) use Close when the event is completed; and (8) append the selected transition to $U_i^{(n+1)}$ and validate uniqueness, evidence monotonicity, temporal consistency, and terminal-state stability.

Link and Update are separated by the location of the new information: an independently retrievable unit remains a fact and is linked, whereas an attribute or evidence item is absorbed by Update. Merge is reserved for equivalent event units; Link preserves related but non-equivalent units. This distinction prevents compaction from erasing causality or provenance.

ILLUSTRATIVE PROCESSING AND PROTOTYPE OBSERVATIONS

Consider a service-request stream in which the first record reports a water-supply failure at a given address. With no active equivalent, the record is inserted. A second complaint from another user, referring to the same location, event type, and time interval, is merged because it describes the same incident. A photograph submitted by an inspection team may remain a separate evidence fact and be linked to the incident.

A later record containing the repair-team identifier and an updated completion time enriches the existing fact through Update. Finally, a resolved status closes the lifecycle. The five transitions therefore describe one incident without turning every incoming record into an independent active fact.

The lifecycle can therefore be represented without five independent active records: the initial incident becomes the main fact; an equivalent complaint is consolidated by Merge; an inspection image or related log remains traceable through Link; new attributes are incorporated by Update; and a resolved status is recorded by Close.

The operator logic was implemented in the FaktRepo-Prototype and FaktAnalytika-Prototype tools and demonstrated on 20,000 normalised NYC 311 service-request records. An append-only representation would keep all 20,000 records as independent units. The operator-controlled repository produced 6,319 final facts, consolidated 13,681 duplicate or near-duplicate records, and preserved 19,544 semantic or evidential relations. The final active representation ratio was

$$Representation\ ratio = \frac{6\,319}{20\,000} \times 100\% = 31.60\%. \quad (7)$$

Thus, only 31.60% of the raw record count remained as final fact units, while the relation layer retained the evidence structure needed for audit and explanation. This result is important because pure deduplication could achieve compaction by discarding repeated records, but it would not preserve why the records were considered related. Conversely, an append-only model would preserve every record but would leave the semantic burden to later queries. The proposed operator distinction performs consolidation at ingestion time and stores the relation or update history together with the fact state.

The main operational risk is an incorrect boundary between Merge and Link. A permissive merge threshold may collapse distinct incidents that happen near the same referent, whereas an overly strict threshold leaves duplicates active. The decision therefore requires domain-specific normalisation of referents and event classes, as well as careful selection of θ_m and θ_t. The audit history U_i reduces this risk because every operator application remains inspectable and can be verified against the original evidence.

CONCLUSION

The operator-selection procedure separates Merge, Link, and Update by semantic identity and the role of incoming information. Merge consolidates equivalent event units, Link preserves related but distinct units, and Update enriches an existing fact. Insert and Close complete the lifecycle. In the prototype demonstration, 20,000 stream records were represented by 6,319 final facts while 19,544 relations remained auditable. The method limits repository growth without sacrificing evidence history and provides a reliable basis for later indexing and inter-agent search.

REFERENCES

[1] G. Weiss (Ed.), Multiagent Systems. MIT Press, 1999.

[2] M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Wiley, 2009.

[3] D.E. Xujamberdiyev and O.A. Mamaraufov, “Fact-centered formal model and repository update rules,” IJDT, vol. 9, no. 1, pp. 117–123, 2026.

[4] D.E. Khujamberdiyev, “Fact-centered organization of digital information flow and incremental repository maintenance,” Science and Innovation, vol. 5, no. 5, pp. 81–88, 2026.

[5] NYC Open Data, “311 Service Requests from 2010 to Present,” 2026.