JOT JOURNAL OF OBJECT TECHNOLOGY

Journal of Object Technology | RESEARCH ARTICLE

Ontology-Grounded Modeling of Dynamic Processes: A Case Study in Conflict Analysis

Bahareh Fatemi*, **Fazle Rabbi***, **Andreas Opdahl***, **Yngve Lamo**†, **and Adrian Rutle**†
*University of Bergen, Norway
†Western Norway University of Applied Sciences, Norway

ABSTRACT
Dynamic processes such as conflict escalation, disease progression, or policy change often involve entities transitioning through structured states over time. Capturing these evolving trajectories in a way that is both formally rigorous and adaptable to unstructured data remains a challenge. In this paper, we present a hybrid approach that integrates formal modeling techniques with large language models (LLMs) to support structured reasoning over dynamic phenomena. Although LLMs offer powerful capabilities for information extraction, their use as standalone analytic tools in sensitive domains is limited, as they are prone to generating unreliable outputs and lack grounding in domain theories. To address this, we propose a framework that combines LLM-based event extraction with ontology-grounded knowledge graph construction, using constraint-preserving graph transformations to ensure semantic and structural validity. We demonstrate the framework in the conflict domain by constructing an ontology based on Glasl's conflict escalation model and using it to guide structured event modeling and trajectory analysis. We implement a prototype system demonstrating the framework in the conflict domain, and present experimental results on an event dataset from ACLED.

KEYWORDS Dynamic Process Modeling, Ontologies, Knowledge Graphs, Graph Transformation, Large Language Models.

1. Introduction

Dynamic processes are central to many domains, from political conflicts and patient health to financial markets. Each evolves through sequences of events that drive transitions between states: a dispute may escalate into violence and later de-escalate through negotiation, a patient may alternate between symptomatic phases, and markets may swing between bullish and bearish cycles. Capturing these dynamics requires more than recording isolated events; it demands representations that trace how events accumulate into recognizable state changes and trajectories over time.

In many domains, the primary evidence of such dynamics comes in the form of unstructured reports such as news event reports, clinical notes, financial bulletins, etc. To turn these descriptions into analyzable trajectories, raw text must be transformed into structured representations. Large language models (LLMs) provide powerful new tools for this transformation (Xu et al. 2024; Braun & Oswald 2025). However, while LLMs are increasingly effective at such information extraction tasks, their use as standalone analytic tools in sensitive domains is problematic (Bommasani 2021). LLMs are inherently limited by their finite context window, preventing them from processing large or temporally extended corpora in a unified manner. This makes it difficult for them to reason over long-running event sequences or maintain consistency across evolving conflict states. Moreover, studies have shown that when LLMs are asked to interpret or analyze conflicts end-to-end, they can generate inconsistent or hallucinatory outputs in military and security contexts (Shrivastava et al. 2024), with the potential to exacerbate escalation risks (Rivera et al. 2024). They also lack grounding in domain theory and offer limited interpretability, making them ill-suited as standalone tools for conflict analysis (Borra et al. 2023; Motta et

JOT reference format:
Bahareh Fatemi, Fazle Rabbi, Andreas Opdahl, Yngve Lamo, and Adrian Rutle. *Ontology-Grounded Modeling of Dynamic Processes::A Case Study in Conflict Analysis*. Journal of Object Technology. Vol. 25, No. 3, 2026.
 http://dx.doi.org/10.5381/jot.2026.25.3.a2

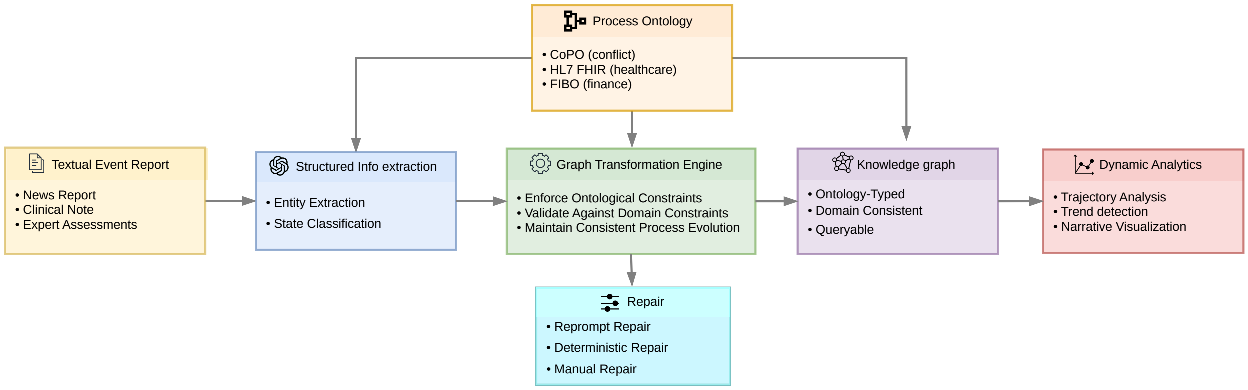

Figure 1 Framework overview: formal methods bridge LLM-based event extraction and knowledge graph construction, enabling structured, ontology-driven dynamic process analysis.

al. 2025). Recent work in safety-critical domains has similarly argued that formal methods can complement the use of LLMs, showing that correctness constraints can be used to detect and iteratively reduce hallucinations (Jha et al. 2023). To this end, we propose a framework that addresses these limitations by combining the scalability of LLM-based event extraction with the rigor and theoretical grounding of symbolic, formally-defined methods.

In this work, we focus on conflict as an inherently dynamic process. In practical settings such as newsrooms or conflict monitoring platforms, analysts need to understand not only isolated incidents but how tensions escalate, stabilize, or de-escalate over time. Supporting such analytical tasks requires representations that capture trajectories rather than static event lists.

Conflict provides an especially suitable domain for this purpose, as it exemplifies the challenges of modeling temporal processes while offering a rich empirical foundation, with decades of systematic event coding and theoretical work on escalation. Prominent event coding systems such as ACLED (Raleigh et al. 2010), UCDP (Sundberg & Melander 2013), and CAMEO (Gerner et al. 2002) have established themselves as indispensable resources for large-scale monitoring of political violence. By categorizing incidents into static taxonomies, they provide broad coverage across countries and time periods. While invaluable for breadth, these systems fall short in representing how disputes evolve through recognizable escalation patterns.

Glasl's nine-stage escalation model (Glasl 1999; Bösch 2017) stands as one of the most influential frameworks for conceptualizing conflict processes (Allwood & Ahlsén 2015). It describes escalation as a progression from latent tensions to destructive confrontation and has been widely applied in mediation and negotiation practice. Despite its explanatory power, Glasl's model has not been operationalized for computational analysis, leaving a disconnect between theory and data-driven research.

We build the Dynamic Process Ontology (DyPO), a schema for modeling evolving processes. Building on it, we introduce the Conflict Process Ontology (CoPO), a machine-operational specialization based on Glasl's model that provides the representational structure for modeling how conflict events accumulate into evolving states over time. Having established this ontology, we integrate it with a constraint-preserving graph transformation mechanism that governs how conflict representations evolve over time within a knowledge graph, which serves as a structured representation of event information. This approach maintains structural validity and enforces key domain specific constraints as new events are incorporated into the knowledge graph. The resulting dynamic graphs support a range of analytic tasks. As an example, we demonstrate their use in conflict trajectory analysis, where evolving representations are structured in accordance with Glasl's escalation theory to capture how conflicts develop over time. Figure 1 illustrates the end-to-end workflow of our framework. An unstructured event report is processed using LLMs to extract structured representations of entities and process states, guided by a process ontology. The extracted information is subsequently integrated into a temporal knowledge graph via a constraint-preserving graph transformation mechanism, which enforces ontological constraints and maintains consistent process evolution. The ontology in this framework provides the semantic schema and constraint definitions that guide both the extraction of structured information and the application of graph transformation rules. It also enables domain-level reasoning by supporting consistency checking and rule-based validation of graph updates. Our contributions are threefold:

- We propose a neuro-symbolic framework for studying dynamic processes by combining the representational power of LLMs with the rigor of symbolic modeling, enabling structured and theory-aligned representations of evolving events.
- We instantiate the framework in the domain of computational conflict analysis by introducing the Conflict Process Ontology (CoPO), formally specified using the Diagram Predicate Framework (DPF) (Rutle 2010) and grounded in Glasl's conflict escalation model.
- We demonstrate the framework through a case study using conflict event data from the ACLED dataset and evaluate the impact of constraint-based refinement under two different repair strategies.

The rest of the paper is organized as follows: Section 2 re-

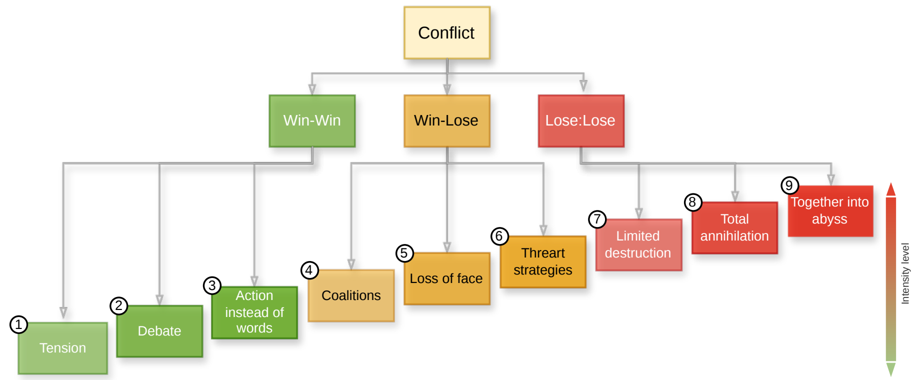

Figure 2 Conflict Process Ontology based on Glasl's conflict model

views related work; Section 3 presents our framework; Section 4 reports results and describes the case study; Section 5 concludes with limitations and future work; Appendix 6 provides prompts and some implementation details.

2. Background

In this section, we review the foundations that our work builds upon. We begin with prior research on ontologies and knowledge graphs in the news and conflict domains, followed by established conflict taxonomies and theories. We then turn to LLMs, describing their potentials and challenges.

2.1. Knowledge graphs

Knowledge Graphs have become an important paradigm for structuring information extracted from unstructured text (Ji et al. 2021; Gallofré Ocaña & Opdahl 2022; Berven et al. 2020). By representing entities and their relationships as nodes and edges, KGs support reasoning, querying, and integration of heterogeneous data sources. In the news domain, prior works have demonstrated the value of KGs for organizing event information, enabling exploratory analysis, and linking events across time and space (Opdahl et al. 2022). For instance, event-centric KGs have been used to connect actors and events across articles, uncover hidden relations, and facilitate downstream tasks such as trend detection or causal inference (Rospocher et al. 2016). In the context of conflict analysis, KGs are particularly attractive because they provide interpretable structures for linking violent events, political actors, and contextual factors (Gastinger et al. n.d.). They allow analysts to trace patterns of interactions, such as alliances or rivalries, and to integrate data from multiple sources. Moreover, by offering explicit semantics, KGs can be integrated with domain ontologies to incorporate structured domain knowledge. Several KG-based approaches have been proposed to capture temporal ordering of events (Yan & Tang 2023; Gottschalk & Demidova 2018), but they lack explicit mechanisms for modeling process state transitions. This limits their ability to represent dynamics such as escalation or remission, where theoretical notions of state progression are essential.

2.2. Conflict Taxonomies

A long tradition in political science and conflict research has focused on systematically coding violent and non-violent events into structured datasets. Prominent conflict event coding systems vary in their scope and granularity. ACLED (Raleigh et al. 2010) which continuously monitors political violence and demonstrations worldwide, distinguishes six top-level event types (Battles, Violence against civilians, Explosions/remote violence, Protests, Riots, Strategic developments) and 25 subevent types. UCDP (Sundberg & Melander 2013) instead organizes events by conflict type, state-based, non-state, and one-sided violence and applies thresholds for fatalities, thereby prioritizing severity and actor type over processual nuance. CAMEO (Gerner et al. 2002) provides the most fine-grained ontology, with over 300 event codes that cover both cooperative and conflictual actions, such as *"Make statement: appeal"* or *"Engage in unconventional mass violence"*. While such taxonomies are valuable for providing breadth across conflicts, their abstraction comes at the cost of depth. They excel at mapping global patterns but do not capture the internal processual dynamics of a given conflict.

2.3. Conflict Escalation Theory

Conflict dynamics have long been studied in psychology and peace research. Among the all contributions is Glasl's nine-stage escalation framework (Glasl 1982), which conceptualizes escalation as a gradual progression from cooperative negotiation to destructive confrontation (Jordan 2000). The nine stages are grouped into three broader levels of escalation, as illustrated in Figure 2. Glasl adopts a broad definition of conflict, enabling his model to capture phenomena ranging from interpersonal disputes to large-scale international crises. Each stage represents a qualitatively distinct phase, beginning with latent tensions and culminating in the most extreme form of escalation, total annihilation. The framework serves a dual role: it is both a diagnostic lens for analyzing the nature and severity of conflict, and a prescriptive roadmap for guiding de-escalation efforts. By situating conflicts within this staged progression, analysts and practitioners can anticipate potential turning points and design more effective strategies for prevention, mitigation, and resolution.

2.4. Large Language Models

LLMs are increasingly used to transform unstructured news into structured event data. Recent surveys highlight a shift toward generative information extraction, demonstrating competitive zero- and few-shot performance across domains (Xu et al. 2024; Stramiglio et al. 2025). A key application is data annotation. Tan et al. review multiple paradigms for using LLMs in textual annotation workflows (Tan et al. 2024), while Ziems et al. show that they can augment annotation in computational social science and reduce reliance on manual coding (Ziems et al. 2024). For example, Elfes (Elfes 2025) applies LLMs to annotate news articles using Greimas' Actantial Model, producing narrative-structured embeddings that capture both semantic and structural aspects of text.

Beyond annotation, LLMs have shown strong performance in news classification. Recent studies report high accuracy for models such as GPT-3.5-turbo and GPT-4o on hierarchical taxonomies like the IPTC news ontology (Fatemi et al. 2023; Kuzman & Ljubešić 2025). Encoder-based transformers such as RoBERTa (Liu et al. 2019) also remain competitive for zero-shot or transfer-style identification of sociopolitical events (Radford 2021). These advances highlight the potential of LLMs for structuring news and conflict data, though their deployment in sensitive domains raises important challenges.

Despite their promise, LLMs outputs are known to suffer from significant limitations, including hallucination, factual inconsistency, limited interpretability, and insufficient reasoning capabilities and domain knowledge (Pan et al. 2024). As a result, their direct application in sensitive domains such as conflict analysis remains problematic (Borra et al. 2023). To mitigate these issues, retrieval-augmented generation (RAG) approaches have been proposed, where LLMs are grounded in external corpora to improve factual reliability (Arslan et al. 2024). Yet, RAG pipelines still struggle with temporal reasoning, especially in conflict analysis, where escalation must be tracked across time rather than retrieved from static corpora (Li et al. 2025). For example, they struggle to answer questions like "Tell me the stage of the conflict in Gaza in the first quarter of 2025". Another widely adopted strategy has been the integration of LLMs with knowledge graphs, which provide structured, interpretable representations of entities and relations (Pan et al. 2024). While KG-augmented LLMs improve grounding and transparency, they often rely on static structures that cannot adequately capture the dynamic, processual nature of conflict. Recently it has been argued that trustworthy AI benefits from the synergistic integration of LLMs and formal methods, with formal reasoning providing rigor and verification while LLMs enhance usability and scalability (Zhang et al. 2024). Jha et al. (Jha et al. 2023) show how formal correctness constraints can be used to detect and iteratively reduce hallucinations in LLM outputs, while Chin et al. Duranti et al. (Duranti et al. 2024) demonstrate how LLMs can be coupled with temporal and description logics through formal verification, and Al Machot et al. (Al Machot et al. 2024) show how ontologies and logical reasoning can be integrated with LLMs to deliver transparent, trustworthy outputs.

Our work extends this line of research by applying the same principle to conflict analysis: LLM-derived annotations are grounded in a formal ontology and validated through constraint-preserving graph transformations, yielding theory-consistent dynamic knowledge graphs.

2.5. Graph Transformation

We introduce only the essentials of graph transformations needed for our approach, without delving into technical depth. For a comprehensive tutorial, see (König et al. 2018).

Graph transformations refer to the process of modifying a graph by applying a sequence of operations including addition or removal of nodes and edges, as well as modifying their properties. A transformation rule is defined as a triple

$$\mathcal{R} = (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R),$$

where L (left-hand side) specifies the pattern to be matched in the current graph, R (right-hand side) specifies the pattern to be constructed, and K (the interface) captures the preserved part of the graph that remains unchanged during the transformation. Intuitively, the rule replaces the occurrence of L by R, while keeping K fixed. In addition to the core rule structure, transformation rules may be equipped with *application conditions AC*. Positive application conditions require that certain patterns or properties must be present in the host graph for the rule to apply. In contrast, negative application conditions specify patterns that must not be present, thereby preventing rule application in situations that would violate well-formedness or domain semantics.

A transformation rule $\mathcal{R}$ specifies the structure and conditions for a graph update. To apply such a rule, we first identify a *match*: a mapping from the left-hand side L of the rule to a subgraph of the host graph G (Ehrig et al. 2015). Intuitively, the match locates where in G the rule can be applied, that is, the part of the graph that corresponds to the pattern described by L. A formal approach for the application of graph transformation rules is the double pushout (Ehrig et al. 1973) framework illustrated in Diagram 1. In this framework, given a match of L as a graph homomorphism m from L into G, a rule application is formalized by first constructing the pushout complement D (i.e., keeping only what should be kept from G), then constructing the pushout H (i.e., adding what should be added into H. This way, applying $\mathcal{R}$ along m yields the transformed graph H.

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "PO"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "n"] \arrow[dr, phantom, "PO"] & R \arrow[d, "m'"] \\
G & D \arrow[l] \arrow[r] & H
\end{tikzcd} \quad (1)$$

In many graphs, nodes and edges are not only connected structurally but also carry *attributes* such as labels, types, or numerical values. To handle this, the double pushout framework is extended to *attributed graph transformations* (König et al. 2018), where transformation rules may additionally update, add, or delete attribute values. This allows us to model changes not only to the graph's topology but also to the descriptive information attached to them.

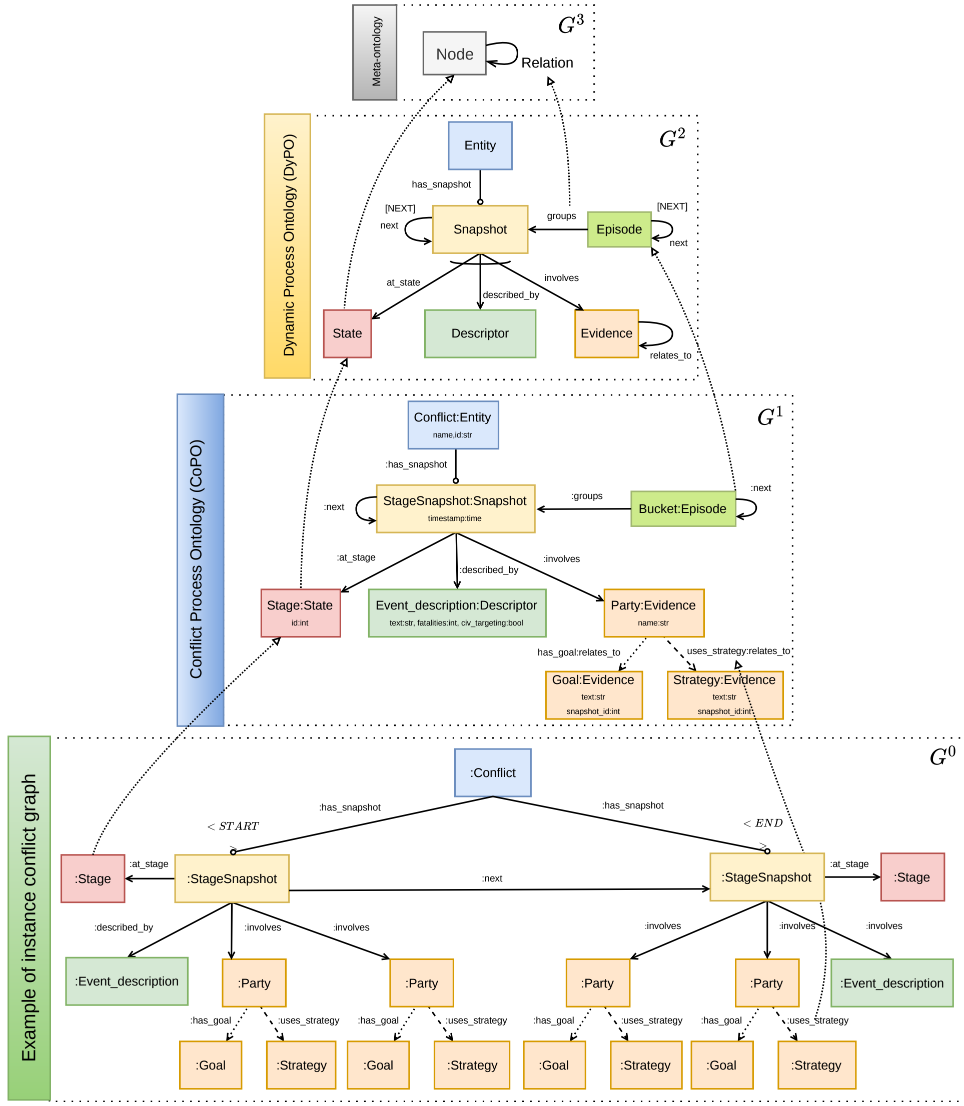

Figure 3 Graph based hierarchy for representing a conflict in the knowledge graph. The semantics of the representation are detailed in Figure 4

3. Modeling of Dynamic Processes

In this section we define the problem setting and describe how constraint-preserving graph transformations enforce structural validity during trajectory evolution of dynamic processing.

3.1. Problem Formulation

We formalize dynamic processes using the *Dynamic Process Ontology (DyPO)*, a schema that provides the core constructs for representing evolving processes. In DyPO, *Snapshots* capture individual events of a process at a given time, including its current *State* and contextual *Evidence*. *Episodes*, provide an intermediate layer of abstraction by grouping multiple snapshots into coherent segments of a process based on desired criteria such as time. Formally, for a process $p \in P$, where P denote the set of dynamic processes (e.g., two concurrent conflicts A and B) modeled in our framework, we define a sequence of evolving graphs:

$$\mathcal{DG}_0^{(p)}, \mathcal{DG}_1^{(p)}, \ldots, \mathcal{DG}_T^{(p)}$$

where $\mathcal{DG}_t^{(p)}$ denotes the state of the process graph after integrating the t-th observation, including the corresponding stage snapshot and its associated contextual evidence. We model the transition from $\mathcal{DG}_{t-1}^{(p)}$ to $\mathcal{DG}_t^{(p)}$ as constraint-preserving graph transformation (Rutle et al. 2010), ensuring that updates remain structurally valid and theory-consistent based on the domain ontology.

In this work, we instantiate DyPO for the domain of conflict analysis, introducing the *Conflict Process Ontology (CoPO)* that encodes the current escalation state of the conflict with respect to Glasl's conflict model. Here, we use the term event report to denote a textual account of a conflict-related occurrence, such as an incident record. Here, an event report denotes a single recorded occurrence (e.g., in ACLED), not multiple news articles about the same event.

Symbol	Shape/Visualization	Semantic Interpretation
$[ConflictHasSnapshots]$	$C \xrightarrow{\texttt{h}} S$	$\forall c \in C: \ \lvert h(c) \rvert \geq 1.$
$[TrajectoryStart]$	$s_1 : S^{<START>} \xrightarrow{\texttt{n}} s_2 : S$	$\forall s \in S : \neg \exists s \ (n(s) = s_1)$
$[TrajectoryNext]$	$S_1 \xrightarrow{\texttt{n [NEXT]}} S_2$	$\forall s_1, s_2 \in S_1 : s_1 \notin n(s_1)$, $n(s_1) = n(s_2) \implies s_1 = s_2$
$[TrajectoryEnd]$	$s_1 : S \xrightarrow{\texttt{n}} s_2 : S^{<END>}$	$\forall s \in S : \neg \exists s \ (n(s_2) = s)$
$[SnapshotStructure]$	$\begin{tikzcd} & S \arrow[dl, "\texttt{a}"'] \arrow[d, "\texttt{d}"] \arrow[dr, "\texttt{i}"] & \\ St & E & P \end{tikzcd}$	$\forall s \in S : \ \lvert a(s) \rvert = 1 \ \wedge \ \lvert d(s) \rvert = 1 \ \wedge \ \lvert i(s) \rvert \geq 1$
$[ActorSemantics]$	$\begin{tikzcd} & P \arrow[dl, dotted, "\texttt{h}"'] \arrow[dr, dashed, "\texttt{u}"] & \\ G & & S \end{tikzcd}$	$\forall p \in P : \ \lvert h(p) \rvert \geq 1 \ \wedge \ \lvert u(p) \rvert \geq 1.$

Figure 4 Signature Σ containing domain specific predicates.

Each event report provides contextual evidence that describes, e.g., the involved parties along with their goals and strategies and contributes a *stagesnapshot* that encodes the current escalation state of the conflict. A conflict $c \in C \subseteq P$ is represented as a sequence of graphs constructed from a set of event reports $\{e_1, e_2, \dots, e_T\}$:

$$\mathcal{CG}_0^{(c)}, \mathcal{CG}_1^{(c)}, \dots, \mathcal{CG}_T^{(c)}$$

Each report e_i is processed into a stage snapshot s_i, which captures the conflict's current state along with its contextual evidence. Each graph $\mathcal{CG}_i^{(c)}$ thus reflects the cumulative evolution of the conflict up to step i, incorporating all snapshots $\{s_1, \dots, s_i\}$ and their associated semantics.

The trajectory of conflict c is then defined as the ordered sequence of stage-labeled snapshots:

$$\tau(c) = \langle (s_1, st(s_1)), (s_2, st(s_2)), \dots, (s_T, st(s_T)) \rangle,$$

which describes how the conflict unfolds over time.

To study events from a more abstract and higher level, snapshots can be aggregated into temporal groups known as *buckets* or *episodes*, where each bucket contains multiple snapshots into a single representative stage. Let $\mathcal{B} = \{b_1, b_2, \dots, b_M\}$ be a segmentation of $\tau(c)$ into episodes, and S denote the set of stage snapshots of conflict c. For each bucket $b \in \mathcal{B}$, let $S_b = \{s \in S \mid \text{bucket}(t(s)) = b\}$ denote the set of snapshots assigned to bucket b based on their timestamp, and let $\hat{st}(b)$ be the aggregated stage label computed from S_b. The *bucketed trajectory* of conflict c is then:

$$\tau_{\mathcal{B}}(c) = \langle (b_1, \hat{st}(b_1)), (b_2, \hat{st}(b_2)), \dots, (b_M, \hat{st}(b_M)) \rangle.$$

3.2. CoPO formalization

In designing CoPO, we opted for a minimal core structure built around actors, goals, strategies, stages and time which are either directly available as event metadata or extracted using an LLM. This reflects how conflicts are commonly narrated in news media: stories often revolve around *who* is involved, *what* they want, *how* they pursue it, *what* and *when* happened, situated within a broader trajectory of escalation or de-escalation. By aligning the ontology with these basic narrative structures as well as escalation semantics coming from Glasl's model, CoPO facilitates general applicability across across diverse conflict domains, including political, organizational, and intergroup conflicts. Although this minimal design provides a flexible foundation, domain-specific extensions can be introduced where needed, without compromising the ontology's broad usability across heterogeneous conflict contexts.

To guarantee structural and semantic validity, we adopt the *Diagram Predicate Framework (DPF)* (Rutle 2010) for formalization. DPF is a meta-modelling framework that represents models as layers of typed graphs: each graph is typed by a higher-level graph. The higher levels provide increasingly abstract descriptions of the levels below. In our case shown in Figure 3, G^3 defines the meta-ontology, G^2 defines the DyPO, G^1 encodes the CoPO, and G^0 corresponds to conflict instances, i.e., the concrete conflict graphs $\mathcal{CG}_t^{(c)}$ in the sequence. Typing morphisms depicted as dotted arrows ensure that every instance graph conforms to the structural rules declared at the ontology level.

To capture well-formedness within the model, we define diagrammatic predicates. A *signature* Σ provides a collection of domain-specific predicates. A predicate specifies a *symbol* together with a *shape* that identifies the structure of the graph on which the predicate is applicable; and an optional visualization

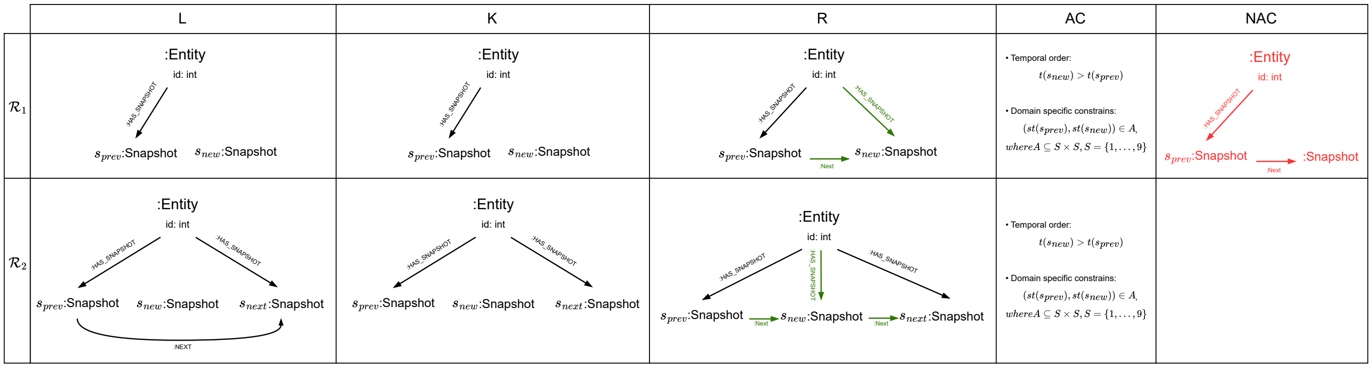

Figure 5 Transformation rules *AddSnapshot* ($\mathcal{R}_1$) and *AddInterSnapshot* ($\mathcal{R}_2$) for adding new snapshot.

of the predicate. The semantic interpretation of each predicate could be defined using various mathematical frameworks, here we use set theory for this purpose. Originally, DPF comes with a set of built-in standard general-purpose predicates such as cardinality bounds, injectivity, surjectivity etc. that capture basic structural properties of graphs. Here we build on these primitives by combining them into domain-specific predicates.

Figure 4 presents a representative set of atomic predicates. It specifies a predicate signature Σ, showing shapes and interpretations. For example, the predicate $[ConflictHasSnapshots]$ is a total mapping in the multi-valued function semantics requiring each `Conflict` node to be linked to at least one corresponding `StageSnapshot` node. Similarly $[SnapshotStructure]$ enforces that each snapshot is associated with exactly one stage by defining a total single-valued mapping, while also requiring links to actors and evidence defining a total multi-valued mapping.

The constraints serve to rigorously regulate graph evolution, ensuring structural consistency throughout the incremental updates associated with conflict dynamics.

3.3. Graph-Based Evolution

The evolution of the knowledge graph is governed through graph transformation rules, which regulate how new events are incorporated. Formally, let $\mathcal{CG}_{t-1}^{(c)}$ denote the conflict graph representing the state of conflict c after integrating the first $t-1$ events. Given a new event report at time step t, the graph is incrementally updated to $\mathcal{CG}_{t}^{(c)}$ by incorporating the newly extracted information. This update is performed through the application of a graph transformation rule, which identifies the relevant context in $\mathcal{CG}_{t-1}^{(c)}$ and extends it to produce the updated graph $\mathcal{CG}_{t}^{(c)}$. Attributed graph transformation rules enable the addition of new stage snapshots, the establishment of temporal relations between them, and the association of supporting evidence, while ensuring that the resulting graph conforms to CoPO and its DPF-based constraints.

The central transformation in our framework is the *AddSnapshot* ($\mathcal{R}_1$) rule shown in Figure 5. This rule introduces a new stage snapshot, together with its associated evidence, into the graph when a new report is received. For simplicity, the evidence is not depicted in the Figure.

To preserve the temporal order an application condition makes sure that the new snapshot s_{new} at timestamp t must have a strictly greater timestamp than the previous snapshot s_{prev} at timestamp $t-1$, i.e. $t(s_{new}) > t(s_{prev})$. This preserves chronological consistency. The second AC imposes domain specific constraints that govern the semantic validity of stage transitions. It verifies whether the transition from the previous stage $st(s_{\text{prev}})$ to the new stage $st(s_{\text{new}})$ is included in a predefined set of admissible transitions $\mathsf{A} \subseteq K \times K$, where $K = \{1, \ldots, 9\}$ represents Glasl's escalation stages. The admissibility relation can be tailored to the domain, for example by incorporating thresholds for the number of fatalities, the targeting of civilians, or the involvement of specific actor strategies etc. The NAC ensures the prior snapshot s_{prev} must not already be connected to another snapshot via a $NEXT$ edge. This prevents branching and enforces linear trajectories. It also automates the application of *AddInterSnapshot* ($\mathcal{R}_2$) rule in cases where an intermediate snapshot is to be added. These conditions constitute a minimal set of constraints. More complex domain-specific requirements can be naturally expressed through additional application conditions, as illustrated in our experiments.

While the AddSnapshot rule appends a new stage snapshot at the end of the conflict trajectory, temporal updates sometimes require inserting information between existing snapshots. For example, a newly discovered report may correspond to an event that occurred earlier than already recorded snapshots, or timestamps may require correction. To handle such cases, we define the AddInterSnapshot rule. This rule rewires the conflict graph so that a new snapshot s_{new} is consistently inserted between an existing predecessor s_{prev} and successor s_{next}. Concretely, the direct edge from s_{prev} to s_{next} is removed and replaced with two edges, $s_{prev} \rightarrow s_{new}$ and $s_{new} \rightarrow s_{next}$, while preserving the temporal continuity and compliance with constraints.

While the application conditions embedded in the transformation rules capture basic structural requirements, additional conditions can be introduced to enforce domain-specific consistency. We incorporate three such constraints: (i) the *violence-consistency floor* (VCF), preventing non-violent stage assignments for events that exhibit violence indicators; (ii) *severity-aware transition smoothing* (SATS), limiting unrealistic stage jumps between consecutive events based on event severity; and (iii) *escalation-shock monotonicity* (ESM), preventing stage decreases following sharp increases in violence. These can be expressed as negative application conditions (NACs) attached to the transformation rules. Formal definitions are provided in Appendix 2. When a NAC detects a violation, the system resolves it through a repair mechanism. Violations may be handled through rejection, re-prompting the LLM, or deterministic (hard-coded) correction.

3.4. Aggregation Through Buckets

The notion of *buckets* in the CoPO provides a flexible mechanism for aggregating snapshots into higher-level units of analysis. Buckets act as containers that group snapshots according to a chosen criterion, allowing processes to be studied at multiple granularities. While the most natural criterion is time, e.g., grouping snapshots by week, month, etc., other domain-specific criteria can be applied. For example, snapshots may be grouped based on stage, actors and other dimensions. By abstracting over individual snapshots, they facilitate stability against noisy observations and serve as summarized representations of events over selected dimensions.

To illustrate the approach, we focus on *temporal bucketing*, which aggregates snapshots based on time intervals and enables the construction of conflict trajectories at different temporal granularities. This is necessary because conflicts often involve multiple reports within a given period, each of which may correspond to different escalation stages. To construct a coherent trajectory, these variations must be reconciled into a single representative stage for each bucket.

Formally, each bucket b contains a set of snapshots S_b, each with an associated stage classification. Within a bucket, snapshots are weighted by their severity, taking into account fatalities and civilian targeting. To determine the representative stage of a bucket, we employ a *soft voting mechanism*: each snapshot contributes not only to its own stage but also to nearby stages, with decaying influence as the stage distance increases. This prevents erratic jumps due to noisy or ambiguous reports and yields trajectories that are both stable and escalation-sensitive. The procedure is detailed in Algorithm 1.

Algorithm 1 Temporal Bucketing Mechanism

Require: Snapshots $S = \{s_1, \dots, s_T\}$ with stage labels $st(s)$, partition $\mathcal{B} = \{b_1, \dots, b_M\}$ at granularity $g \in \{\text{week}, \text{month}, \text{quarter}\}$, parameters α (decay rate), λ_1 (fatalities weight), λ_2 (civilian-targeting weight), β (bias), stage set $K = \{1, \dots, 9\}$
Ensure: Bucketed trajectory $\tau_{\mathcal{B}}(c)$

for each bucket $b \in \mathcal{B}$ **do**
  $S_b \leftarrow \{s \in S \mid \text{bucket}(t(s), g) = b\}$
  for each $s \in S_b$ **do**
    $\text{sev}(s) \leftarrow \lambda_1 \cdot \log(1 + \text{fat}(s)) + \lambda_2 \cdot \text{ct}(s) + \beta$ ▷ $\text{fat}(s)$: fatalities in snapshot s; $\text{ct}(s)$: civilian-targeting indicator in snapshot s
  for $k \in K$ **do**
    $\text{score}_b(k) \leftarrow \sum_{s \in S_b} \text{sev}(s) \cdot \exp(-\alpha \cdot \|st(s) - k\|)$
  $\hat{st}(b) \leftarrow \arg\max_{k \in K} \text{score}_b(k)$
return $\tau_{\mathcal{B}}(c) = \langle (b_1, \hat{st}(b_1)), \dots, (b_M, \hat{st}(b_M)) \rangle$

4. Experiments

As the CoPO is defined as an attributed graph model, we adopt Neo4j as the underlying graph database. Neo4j natively supports labeled property graphs, which map directly to our representation of nodes and their attributes. Furthermore, Neo4j's query language Cypher provides the expressive power we need to implement graph transformation rules, such as the insertion of new snapshots, the linking of parties, and the enforcement of structural conditions. This choice enables both efficient storage of large-scale conflict data and operationalization of our rule-based updates within a widely used graph database system.

4.1. Event Annotation

As Glasl's conflict ontology has not previously been operationalized in a computational setting, no established benchmarks are available. In this study, for evaluation and demonstration purposes, we collect 1,000 events related to the Israel–Palestine conflict in 2023 from the ACLED dataset. A subset of 100 events is manually annotated with escalation stage labels by two human annotators, and these annotations are used to evaluate the predictions of several LLMs in a zero-shot setting. Figure 9 in the appendix illustrates the prompt structure and a sample response of the LLM on an input ACLED event. To guide the LLM, we constructed a list of key indicators based on Glasl's definitions and included these in the prompt design as stage description and key indicators which can be found on Zenodo (Fatemi 2026).

Annotation performance was assessed using mean absolute error (MAE), tolerant accuracy (within ± 1 stage), and quadratic weighted Cohen's Kappa (κ_w). These metrics capture complementary aspects of prediction quality in our ordinal setting.

- **Mean Absolute Error (MAE):** measures the average absolute difference between predicted stage labels $\hat{y}_i$ and true labels y_i across n events:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

  Here, MAE quantifies the magnitude of prediction errors.
- **Tolerant Accuracy (Acc. (± 1)):** measures the proportion of predictions within ± 1 category of the true label:

$$\text{Acc.}(\pm 1) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(|\hat{y}_i - y_i| \leq 1)$$

  where $\mathbf{1}(\cdot)$ is the indicator function. This metric captures how often we are close enough for stage-level interpretation, given the inherent subjectivity of stage annotation.

  While MAE and tolerant accuracy provide intuitive measures of error magnitude and near-correctness, they do not

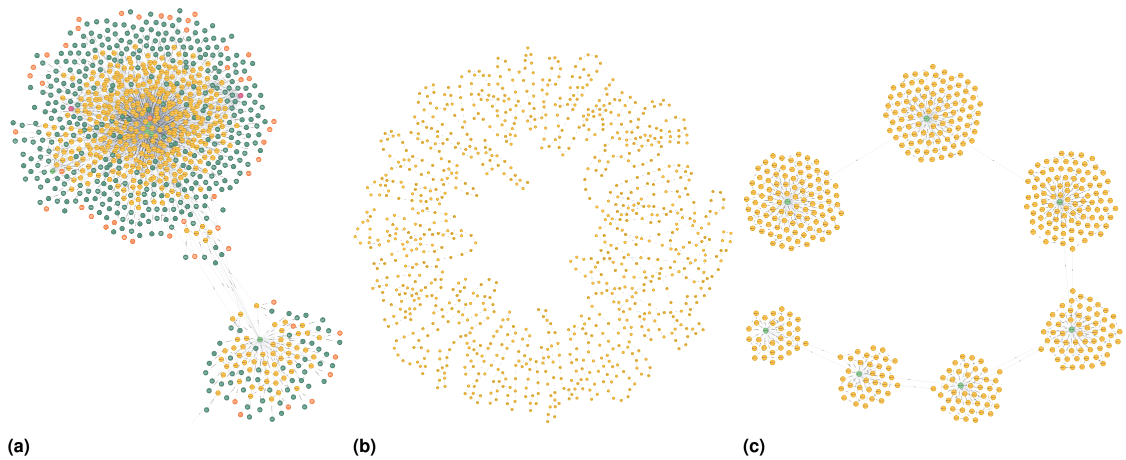

Figure 6 Israel–Palestine conflict knowledge graph at different aggregation levels: (a) full event-level graph, (b) ungrouped conflict trajectory, and (c) weekly bucketed trajectory showing the evolution of escalation over time. Node types are color-coded as in Figure 3.

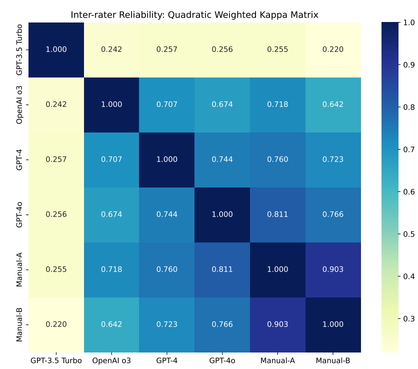

Figure 7 Quadratic weighted Cohen's Kappa matrix showing agreement between annotation methods

account for agreement expected by chance and treat all errors uniformly. In particular, they do not fully capture the ordinal structure of conflict stages, where larger deviations should be penalized more strongly.

– **Quadratic Weighted Cohen's Kappa (κ_w).** Measures agreement with the reference beyond chance, with a heavier penalty for larger disagreements. For k categories, the weight matrix is defined as

$$w_{i,j} = \left(\frac{i-j}{k-1}\right)^2, \qquad \kappa_w = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

for $i, j \in \{1, \ldots, k\}$, where O is the observed agreement matrix and E is the expected agreement matrix under independence.

To assess agreement between LLM and human annotations, we compute pairwise quadratic weighted Cohen's kappa scores, shown in Fig. 7. Each event is annotated by four OpenAI models and two human annotators serving as references. The heatmap shows strong agreement between the human annotators ($\kappa_w = 0.903$). Among the models, GPT-4o achieves the highest agreement with human annotations.

Building on this analysis, we investigate whether the application conditions introduced in Section 3.3 improve the quality of LLM-based annotations. Specifically, we apply the three conditions defined earlier: VCF, SATS and ESM. As refine strategy, violations are resolved through *deterministic repair* and *reprompt-based repair*, condition-aware prompting the LLM to revise its prediction.

Using GPT-4o annotations as the base, we evaluate the refined outputs against human annotators (Table 1). Deterministic repair consistently improves performance, with SATS yielding the largest gains, indicating that many errors arise from implausible temporal jumps. VCF provides additional semantic corrections, while ESM shows limited standalone impact but contributes when combined with other constraints.

Reprompt-based repair further improves results across all configurations, outperforming deterministic repair. The combination of VCF and SATS achieves the best performance, reaching a weighted kappa of 0.875 and tolerant accuracy of 0.960 (Manual-A), and 0.845 and 0.940 (Manual-B). Overall, constraint-aware refinement improves both semantic consistency and temporal plausibility of LLM predictions.

4.2. Prototype

As a proof of concept, we implemented the framework in the Neo4j graph database. Neo4j supports attributed property graphs and the Cypher query language, enabling concise implementation of the graph transformation rules while efficiently handling large, evolving knowledge graphs. The full Cypher query for the AddSnapshot rule is provided in the Appendix.

In our event dataset, while information such as involved actors, number of fatalities and civilian targeting are already provided in ACLED, we extracted conflict stages, goals, and strategies in a zero-shot setting.

A knowledge graph is built incrementally based on the extracted information which links conflicts to stage snapshots, along with corresponding evidence. Figure 6a shows a fragment of the Neo4j database incrementally created by applying the

Table 1 Performance of LLMs and rule-based refinement across two human references, including reprompt-based repair.

Reference: Manual-A			
Method	MAE ↓	κ_w ↑	Acc. (± 1) ↑
Base models			
GPT-3.5	1.830	0.255	0.310
O3	0.520	0.721	0.900
GPT-4	0.460	0.760	0.890
GPT-4o	0.310	0.811	0.930
rule-based refinement (deterministic repair)			
+ VCF	0.290	0.837	0.930
+ SATS	0.290	0.847	0.940
+ ESM	0.280	0.835	0.940
+ VCF + SATS + ESM	0.280	0.851	0.940
rule-based refinement (reprompt repair)			
+ VCF (reprompt)	0.290	0.837	0.930
+ SATS (reprompt)	0.260	0.865	0.950
+ ESM (reprompt)	0.290	0.832	0.940
+ VCF + SATS + ESM (reprompt)	**0.250**	**0.875**	**0.960**

Reference: Manual-B			
Method	MAE ↓	κ_w ↑	Acc. (± 1) ↑
Base models			
GPT-3.5	2.040	0.220	0.230
O3	0.670	0.645	0.840
GPT-4	0.490	0.723	0.860
GPT-4o	0.380	0.766	0.910
rule-based refinement (deterministic repair)			
+ VCF	0.360	0.790	0.910
+ SATS	0.340	0.825	0.930
+ ESM	0.350	0.788	0.920
+ VCF + SATS + ESM	0.330	0.828	0.930
rule-based refinement (reprompt repair)			
+ VCF (reprompt)	0.360	0.790	0.910
+ SATS (reprompt)	0.310	0.836	0.930
+ ESM (reprompt)	0.360	0.785	0.920
+ VCF + SATS + ESM (reprompt)	**0.300**	**0.845**	**0.940**

ADDSNAPSHOT rule across reports to produce a sequence of stage snapshots (yellow nodes) connected by `NEXT` relations. The fine-grained conflict trajectory or escalation path for Israel in 2023 (Figure 6b) illustrates this linear chain of snapshots, which quickly becomes dense given the high event frequency. The aggregation through buckets allows for grouping these events in degrees of granularity. Figure 6c shows seven clusters of snapshots representing seven weekly buckets shown as green nodes.

Figure 8a presents the per-event trajectories without bucketing, illustrating the noisiness. To obtain interpretable trajectories, we apply the severity-weighted bucketing mechanism described in Section 3.4 with parameters $\alpha = 0.6$, $\lambda_1 = 0.8$, $\lambda_2 = 0.4$, and $\beta = 0.2$, aggregating events at weekly and monthly granularities. The resulting trajectories are shown in Figures 8b and 8c.

These parameters represent configurable weights that can be adjusted by the user or domain experts to reflect domain-specific

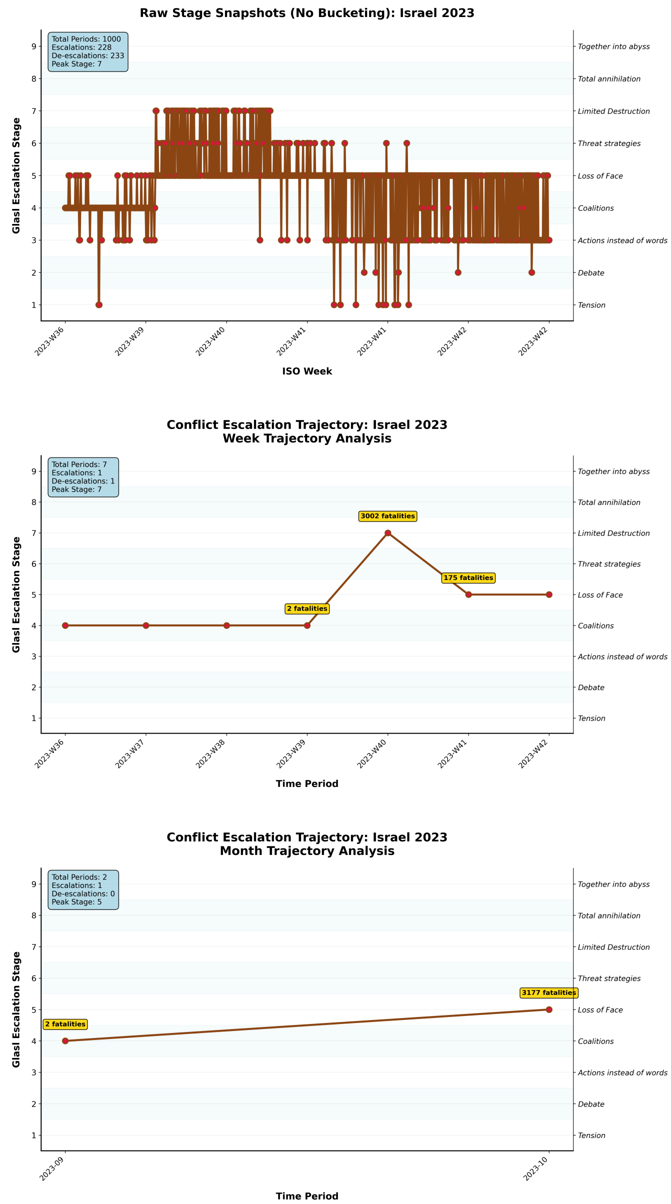

Figure 8 Weekly and monthly trajectories of the Israel–Palestine conflict.

priorities, and the values used here serve as an illustrative configuration. The peak on the 2023-W40 reflects the sharp escalation of the conflict in October 7th, with 3002 casualties reported by ACLED. The choice of granularity depends on the analytical goal, as while raw event-level snapshots capture individual occurrences, aggregated bucketed trajectories smooth over noise from individual reports, providing a coherent view of the overall dynamics of escalation necessary for tasks like trend detection.

5. Conclusion and Discussion

In this work, we have introduced a hybrid framework that integrates LLMs with formal methods to support structured reasoning over dynamic processes extracted from unstructured text.

While LLMs offer impressive capabilities for contextual understanding, we identified limitations in their ability to produce more reliable, theory-consistent and structured representations. To bridge this gap, our framework combines LLM-based information extraction with ontology-driven graph construction, using constraint-preserving graph transformation to ensure semantic and structural validity.

We instantiated this framework in the conflict domain by developing a machine-interpretable ontology grounded in Glasl's nine-stage escalation theory. The resulting system supports the extraction, structuring, and temporal analysis of conflict trajectories from real-world news data. Empirical results and a prototype implementation demonstrates the feasibility of the approach and highlights the potential of combining LLMs with formal modeling techniques for a more interpretable dynamic process analysis.

Looking forward, several directions offer opportunities for extension. First, the framework can be applied to other domains involving structured process evolution, such as disease progression, financial market cycles, or crisis response. Second, the integration of real-time or streaming textual inputs may support dynamic updates. Furthermore, the presented evaluation focuses on demonstrating the effect of integrating formal constraints with LLM-based predictions in a realistic conflict setting. The selected events are drawn from a concentrated period of the conflict, providing a suitable context for observing the behavior of the proposed constraints. Future work may explore broader datasets covering a wider range of conflict stages, as well as additional expert annotations, to further validate and extend these findings. Finally, improving the controllability and reliability of LLM outputs through prompt engineering, fine-tuning remains a promising complement to the ontology-driven approach proposed here.

References

Allwood, J., & Ahlsén, E. (2015). On stages of conflict escalation. In *Conflict and multimodal communication: Social research and machine intelligence* (pp. 53–69). Springer.

Al Machot, F., Horsch, M. T., & Ullah, H. (2024). Building trustworthy ai: transparent ai systems via language models, ontologies, and logical reasoning (transpnet). In *Designing the conceptual landscape for a xair validation infrastructure workshop* (pp. 25–34).

Arslan, M., Munawar, S., & Cruz, C. (2024). Political-rag: using generative ai to extract political information from media content. *Journal of Information Technology & Politics*, 1–16.

Berven, A., Christensen, O. A., Moldeklev, S., Opdahl, A. L., & Villanger, K. J. (2020). A knowledge-graph platform for newsrooms. *Computers in Industry*, *123*, 103321.

Bommasani, R. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Borra, E., Bollini, P., Caroleo, L., Cerigo, B., Dubèl, R., Gaw, F., ... Sombatpoonsiri, J. (2023). *A field guide to using llms for online conflict analysis.* https://www.digitalmethods.net/Dmi/FieldGuideToUsingLLMs. (Digital Methods Initiative, University of Amsterdam. Accessed: 2025-09-10)

Braun, L., & Oswald, C. (2025). Automated information extraction from text variables in event datasets with large language models.

Bösch, R. (2017, 11). *Conflict escalation.* Oxford University Press. Retrieved from https://oxfordre.com/internationalstudies/view/10.1093/acrefore/9780190846626.001.0001/acrefore-9780190846626-e-82 doi: 10.1093/acrefore/9780190846626.013.82

Duranti, D., Giorgini, P., Mazzullo, A., Robol, M., & Roveri, M. (2024). Llm-driven knowledge extraction in temporal and description logics. In *International conference on knowledge engineering and knowledge management* (pp. 190–208).

Ehrig, H., Ermel, C., Golas, U., & Hermann, F. (2015). Graph and model transformation. *Monographs in Theoretical Computer Science*.

Ehrig, H., Pfender, M., & Schneider, H. J. (1973). Graph-grammars: An algebraic approach. In *14th annual symposium on switching and automata theory (swat 1973)* (pp. 167–180).

Elfes, J. (2025). Mapping news narratives using llms and narrative-structured text embeddings. In *Proceedings of the 17th acm web science conference 2025* (p. 326–337). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3717867.3717868 doi: 10.1145/3717867.3717868

Fatemi, B. (2026, March). *Conflict stages and their indicators adopted from glasl's model.* Zenodo. Retrieved from https://zenodo.org/records/18832872 doi: 10.5281/zenodo.18832872

Fatemi, B., Rabbi, F., & Opdahl, A. L. (2023). Evaluating the effectiveness of gpt large language model for news classification in the iptc news ontology. *IEEE Access*, *11*, 145386-145394. doi: 10.1109/ACCESS.2023.3345414

Gallofré Ocaña, M., & Opdahl, A. L. (2022). Supporting newsrooms with journalistic knowledge graph platforms: Current state and future directions. *Technologies*, *10*(3), 68.

Gastinger, J., Sztyler, T., Steinert, N., Gründer-Fahrer, S., Martin, M., Schuelke, A., & Stuckenschmidt, H. (n.d.). Dynamic representations of global crises: A temporal knowledge graph for conflicts, trade and value networks. In *The third learning on graphs conference.*

Gerner, D. J., Schrodt, P. A., Yilmaz, O., & Abu-Jabr, R. (2002). Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions. *International Studies Association, New Orleans*.

Glasl, F. (1982). The process of conflict escalation and roles of third parties. In *Conflict management and industrial relations* (pp. 119–140). Dordrecht: Springer Netherlands. doi: 10.1007/978-94-017-1132-6_6

Glasl, F. (1999). *Confronting conflict: a first-aid kit for handling conflict*. Hawthorn press.

Gottschalk, S., & Demidova, E. (2018). Eventkg: A multilingual event-centric temporal knowledge graph. In *European semantic web conference* (pp. 272–287).

Jha, S., Jha, S. K., Lincoln, P., Bastian, N. D., Velasquez, A., & Neema, S. (2023). Dehallucinating large language models using formal methods guided iterative prompting. In *2023*

ieee international conference on assured autonomy (icaa) (pp. 149–152).
Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, *33*(2), 494–514.
Jordan, T. (2000). Glasl's nine stage model of conflict escalation. *Available via Mediate. com: http://www. mediate. com/articles/jordan. cfm (Accessed: March 2005)*.
König, B., Nolte, D., Padberg, J., & Rensink, A. (2018). A tutorial on graph transformation. *Graph Transformation, Specifications, and Nets: In Memory of Hartmut Ehrig*, 83–104.
Kuzman, T., & Ljubešić, N. (2025). Llm teacher-student framework for text classification with no manually annotated data: A case study in iptc news topic classification. *IEEE Access*.
Li, D., Niu, Y., Ai, Y., Zou, X., Qi, B., & Liu, J. (2025). T-grag: A dynamic graphrag framework for resolving temporal conflicts and redundancy in knowledge retrieval. *arXiv preprint arXiv:2508.01680*.
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
Motta, E., Daga, E., Gangemi, A., Gjelsvik, M. L., Osborne, F., & Salatino, A. (2025). The epistemology of fine-grained news classification. *Semantic Web*, *16*(3), 22104968251344461. Retrieved from https://doi.org/10.1177/22104968251344461 doi: 10.1177/22104968251344461
Opdahl, A. L., Al-Moslmi, T., Dang-Nguyen, D.-T., Gallofré Ocaña, M., Tessem, B., & Veres, C. (2022). Semantic knowledge graphs for the news: A review. *ACM Computing Surveys*, *55*(7), 1–38.
Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.
Radford, B. J. (2021). Case 2021 task 2: zero-shot classification of fine-grained sociopolitical events with transformer models. In *Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (case 2021)* (pp. 203–207).
Raleigh, C., Linke, R., Hegre, H., & Karlsen, J. (2010). Introducing acled: An armed conflict location and event dataset. *Journal of peace research*, *47*(5), 651–660.
Rivera, J.-P., Mukobi, G., Reuel, A., Lamparth, M., Smith, C., & Schneider, J. (2024). Escalation risks from language models in military and diplomatic decision-making. In *Proceedings of the 2024 acm conference on fairness, accountability, and transparency* (pp. 836–898).
Rospocher, M., Van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., ... Bogaard, T. (2016). Building event-centric knowledge graphs from news. *Journal of Web Semantics*, *37*, 132–151.
Rutle, A. (2010). *Diagram predicate framework: a formal approach to mde* (PhD thesis). The University of Bergen.
Rutle, A., Rossini, A., Lamo, Y., & Wolter, U. (2010). A formalisation of constraint-aware model transformations. In *International conference on fundamental approaches to software engineering* (pp. 13–28).
Shrivastava, A., Hullman, J., & Lamparth, M. (2024). Measuring free-form decision-making inconsistency of language models in military crisis simulations. *arXiv preprint arXiv:2410.13204*.
Stramiglio, A., Schimmenti, A., Pasqual, V., van Erp, M., Sovrano, F., & Vitali, F. (2025). Explicit vs. implicit biographies: Evaluating and adapting llm information extraction on wikidata-derived texts. *arXiv preprint arXiv:2509.14943*.
Sundberg, R., & Melander, E. (2013). Introducing the ucdp georeferenced event dataset. *Journal of peace research*, *50*(4), 523–532.
Tan, Z., Beigi, A., Wang, S., Guo, R., Bhattacharjee, A., Jiang, B., ... Liu, H. (2024). Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.
Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., ... Chen, E. (2024). Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, *18*(6), 186357.
Yan, Z., & Tang, X. (2023). Narrative graph: Telling evolving stories based on event-centric temporal knowledge graph. *Journal of Systems Science and Systems Engineering*, *32*(2), 206–221.
Zhang, Y., Cai, Y., Zuo, X., Luan, X., Wang, K., Hou, Z., ... others (2024). The fusion of large language models and formal methods for trustworthy ai agents: A roadmap. *arXiv preprint arXiv:2412.06512*.
Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can large language models transform computational social science? *Computational Linguistics*, 1–55.

About the authors

Bahareh Fatemi is a PhD candidate at the department of information science and media studies, university of Bergen. You can contact the author at bahareh.fatemi@uib.no.

Fazle Rabbi is associate professor at the department of information science and media studies, university of Bergen. You can contact the author at fazle.rabbi@uib.no.

Andreas Opdahl is professor at the department of information science and media studies, university of Bergen. You can contact the author at andreas.opdahl@uib.no.

Yngve Lamo is professor at the department of Computer science, Electrical engineering and Mathematical sciences, Western Norway University of Applied Sciences. You can contact the author at yla@hvl.no.

Adrian Rutle is professor at the department of Computer science, Electrical engineering and Mathematical sciences, Western Norway University of Applied Sciences. You can contact the author at aru@hvl.no.

6. Appendix

```
<OBJECTIVE>
You are an expert conflict analyst specializing in Glasls nine stage conflict escalation model.
Your task is to classify a single conflict event into the most appropriate escalation stage,
and extract structured information about the actors involved.
You must base your decision strictly on the provided event description and the formal
definitions of Glasls stages.
</OBJECTIVE>

<INSTRUCTIONS>
Follow these steps carefully:
1. Analyze the event description and identify key signals of conflict behavior
   (e.g., cooperation, tension, threats, coercion, violence, destruction).
2. Compare these signals against the definitions and indicators of Glasls stages.
...

Do not rely on external knowledge. Use only the given text and the stage definitions.
</INSTRUCTIONS>

<GUARDRAILS>
- Output ONLY valid JSON (no extra text).
- Stage must be an integer from 1 to 9.
...
</GUARDRAILS>

<CONTEXT>
Event description:
{Event description}

Glasl Conflict Escalation Model:
Level 1: WinWin
Stage 1: Tension
Description: {Stage description}
Indicators: {key indicators}
...
</CONTEXT>

<OUTPUT_FORMAT>
Return the result in the following JSON format:
...
</OUTPUT_FORMAT>
```

```
Event record (input):
"Event description": "On 6 February 2024, Palestinian rioters clashed with Israeli forces in Qaffin (
    Tulkarm, West Bank). Israeli forces opened fire at the rioters, injuring a rioter."

\textbf{LLM response} (output):
{
  "Stage": 5,
  "Actors": [
    {
      "Name": "Palestinian rioters",
      "Strategy": "Clashing with security forces",
      "Goal": "To confront and resist Israeli forces in response to perceived grievances."
    },
    {
      "Name": "Israeli forces",
      "Strategy": "Use of live fire for crowd control",
      "Goal": "To disperse rioters and restore control over the area."
    }
  ]
}
```

Figure 9 Prompt template (top) and corresponding example (bottom) showing the structured event extraction and stage classification for the Israel–Palestine conflict.

NAC	Formal definition	Violation condition
$\mathrm{NAC}_{\mathrm{VCF}}(s_{\mathrm{new}})$	$(fat(s_{\mathrm{new}}) > 0 \lor ct(s_{\mathrm{new}}) = 1) \land st(s_{\mathrm{new}}) < 5$	The new event involves violence, but is assigned a stage below 5.
$\mathrm{NAC}_{\mathrm{SATS}}(s_{\mathrm{prev}}, s_{\mathrm{new}})$	$\lvert st(s_{\mathrm{new}}) - st(s_{\mathrm{prev}}) \rvert > \Delta(s_{\mathrm{new}})$	The stage change between consecutive events exceeds the allowed severity-based jump.
$\mathrm{NAC}_{\mathrm{ESM}}(s_{\mathrm{prev}}, s_{\mathrm{new}})$	$sh(s_{\mathrm{new}}) = 1 \land st(s_{\mathrm{new}}) < st(s_{\mathrm{prev}})$	A shock event defined by a significant increase in fatalities or the onset of civilian targeting occurs, but the stage decreases relative to the previous event.

Table 2 Formal definitions of NACs. $fat(\cdot)$ denotes the number of fatalities, $ct(\cdot) \in \{0,1\}$ indicates civilian targeting, $sh(\cdot) \in \{0,1\}$ denotes whether an event constitutes a shock, and $\Delta(\cdot)$ represents the severity-based maximum allowed stage change.

```
# --- AddSnapshot(c, s_t) ---
# Params expected:
# \$conflict_id, \$ts (ISO string), \$stage (1--9),
# \$event_id, \$event_text, \$fatalities?, \$civ_targeting?,
# \$parties = [{name, goals?:[], strategies?:[]}, ...]

# 1) Conflict + raw snapshot (typed)
MATCH (c:Conflict {id:\$conflict_id})
MERGE (snap:StageSnapshot {
  conflict_id: \$conflict_id,
  timestamp:   \$ts,
  event_id:    \$event_id
})
ON CREATE SET snap.created_at = timestamp()
SET  snap.stage = \$stage
MERGE (c)-[:HAS_SNAPSHOT]->(snap)
MERGE (st:Stage {id:\$stage})
MERGE (snap)-[:AT_STAGE]->(st)

# 2) Evidence (event description)
MERGE (e:Event_Description {id:\$event_id})
SET e.text          = \$event_text,
    e.timestamp     = \$ts,
    e.fatalities    = coalesce(\$fatalities, 0),
    e.civ_targeting = coalesce(\$civ_targeting, false)
MERGE (snap)-[:DESCRIBED_BY]->(e)

# 3) Actors, goals, strategies
WITH c, snap, \$parties AS parties
FOREACH (p IN parties |
  MERGE (pt:Party {name:p.name})
  MERGE (snap)-[:INVOLVES]->(pt)
  FOREACH (g IN coalesce(p.goals, []) |
    MERGE (goal:Goal {text:g})
    MERGE (pt)-[:HAS_GOAL]->(goal)
  )
  FOREACH (stg IN coalesce(p.strategies, []) |
    MERGE (str:Strategy {text:stg})
    MERGE (pt)-[:USES_STRATEGY]->(str)
  )
)

# 4) Temporal link: wire NEXT from latest prior snapshot,
#    but only if that prior snapshot has no successor (no branching)
WITH c, snap
OPTIONAL MATCH (c)-[:HAS_SNAPSHOT]->(prev:StageSnapshot)
WHERE prev.timestamp < \$ts
WITH snap, prev ORDER BY prev.timestamp DESC LIMIT 1
WHERE prev IS NOT NULL AND NOT (prev)-[:NEXT]->()
MERGE (prev)-[:NEXT]->(snap);
```

Figure 10 Cypher query for implementation of `AddSnapshot` rule.