© Volume 2, Issue 10, October 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

World Conflict Map A Real-Time Global Conflict Monitor

[1]Sayali Gaikwad, [2]Vedant Chinchane, [3]Devendra Choudhary [4]Vaishnavi Mhaske
[5]Sakshi Koke

[1]Professor, Department of Information Technology, Parvatibai Genba Moze College of Engineering, Wagholi, Pune, India

[2,3,4,5]Department of Information Technology, Parvatibai Genba Moze College of Engineering, Wagholi, Pune, India

***Abstract*—Global geopolitical events are reported continuously through news organizations and other information sources, making it difficult to understand their geographic distribution and changing patterns from isolated articles or conventional lists. This paper presents WORLD CONFLICT MAP, a web-based real-time/frequent-update global conflict monitoring dashboard that combines GDELT event information with an interactive world map and chronological conflict feed. The system follows a Fetch, Contextualize, Parse, Display workflow in which raw event information is retrieved from GDELT, contextual information is incorporated where available, actors, locations and event categories are extracted through application-level parsing heuristics, and structured event objects are rendered through a responsive web interface. The proposed application uses Next.js, React, TypeScript, Tailwind CSS, React Simple Maps, D3 Scale and Axios. Major features include geographic event markers, regional classification, active-zone highlighting, contextual event popups, source links and a chronological feed. The system is intended as an educational and research-oriented visualization layer rather than an authoritative source of ground truth. The paper describes the architecture, methodology, implementation approach, evaluation protocol, limitations and future scope of the proposed system.**

***Index Terms*—Global conflict monitoring, GDELT, geopolitical events, geographic visualization, Next.js, React, data visualization, real-time/frequent-update dashboard**

I. INTRODUCTION

Geopolitical events are reported continuously by news organizations around the world. The volume, geographic distribution and changing nature of these reports make it difficult for users to understand global conflict activity from isolated articles or conventional lists. Text-based reporting also makes it difficult to perceive relationships between an event and its geographic location. A

map-based interface can reduce this cognitive load by linking event information to locations and presenting related context in a single visual environment.

WORLD CONFLICT MAP addresses this visualization problem by combining event-oriented data from the Global Database of Events, Language, and Tone (GDELT) with an interactive world map. The project emphasizes rapid visual identification of regions, conflict categories and active hotspots. Continental color coding provides a geographic reference layer, while event markers, highlighting, contextual popups and a chronological feed provide additional detail.

The project is implemented as a Next.js application using the App Router structure. React supports component-based interface development, TypeScript provides static typing, Tailwind CSS supports styling, Framer Motion supports interface animation, Lucide React provides icons, React Simple Maps provides SVG-based map rendering, and D3 Scale provides mathematical mappings for visual encodings. The application is intended to demonstrate how public data, API integration, web application development and geographic visualization can be combined into a single information system.

The main contributions of the project are: (i) a unified geographic visualization layer for distributed geopolitical event information; (ii) a structured Fetch, Contextualize, Parse, Display processing pipeline; (iii) event-level contextual popups and direct source links; (iv) a chronological conflict feed complementary to the map; and (v) a responsive interface designed for desktop and mobile use.

The remainder of this paper is organized as follows. Section II reviews related technologies and data sources. Section III presents the system architecture. Section IV describes the data-processing methodology. Section V covers implementation. Section VI presents the evaluation protocol and expected observations. Section VII discusses limitations. Section VIII concludes the paper and outlines future work.

II. RELATED WORK

2.1 GDELT Event Database

The GDELT Project provides a global structured record of physical activities reported in worldwide news, including geographic information and frequent updates. GDELT 2.0 uses the CAMEO event taxonomy and provides georeferencing fields, making it suitable as an input source for event-oriented geographic visualization. The WORLD CONFLICT MAP project uses GDELT as its principal event source and preserves links to original reporting for traceability [1].

2.2 GDELT 2.0 Data Architecture

GDELT documentation describes event and mention records and provides mechanisms for working with current event streams [2]. This supports the project's Fetch, Parse, Display workflow while also emphasizing the importance of source provenance and awareness of event-data limitations. Because the information is derived from news reporting, records may be incomplete, duplicated, delayed or incorrectly interpreted.

2.3 Web and Geospatial Visualization Technologies

Next.js provides the application framework and App Router architecture [3], while React enables reusable component-based user interfaces [4]. React Simple Maps provides declarative SVG map components based on geographic data and projections [7]. D3 Scale provides mappings from abstract data values to visual representations such as position, color, stroke width and symbol size [8]. Together, these technologies support the interactive geographic visualization required by the project.

2.4 Research Gap

Many information systems expose geopolitical news as articles or lists, while separate visualization systems may focus on maps without preserving source context. WORLD CONFLICT MAP combines a frequently updated event feed, geographic visualization, contextual details and source links within one responsive web interface. The project therefore demonstrates an integrated visualization layer rather than claiming to replace established event databases or professional intelligence systems.

III. SYSTEM ARCHITECTURE

3.1 Overview

The proposed system follows a layered architecture. GDELT V2 event endpoints provide global event information. The application data layer retrieves raw event logs and applies contextualization and parsing heuristics. The processed information is converted into consistent event objects containing fields such as event ID, source, event type, location, actors, intensity and timestamp. Next.js and React then render these objects through the map, feed and contextual detail components.

The overall data flow is: GDELT, Fetch, Contextualize, Parse, Event Objects, Map/Feed, User (Fig. 1).

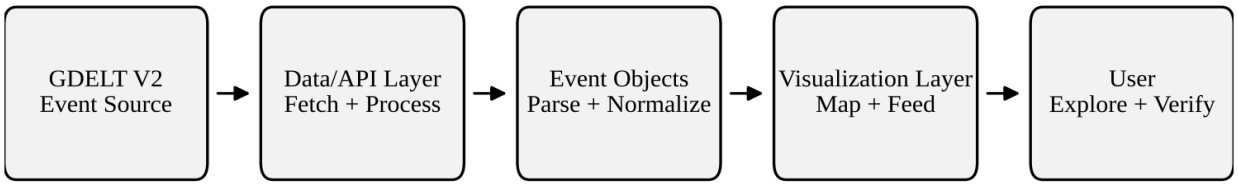

Fig. 1. Conceptual system architecture of WORLD CONFLICT MAP.

3.2 Data Source Layer

GDELT V2 event endpoints provide the principal event information. The project retrieves event logs and uses their geographic and event-related fields as the basis for application-level visualization.

3.3 Data/API Layer

The application retrieves raw event logs, contextualizes selected information and applies parsing heuristics to extract or normalize actors, geographic locations and event types. This layer prepares structured event objects for the presentation layer.

3.4 Visualization Layer

React Simple Maps renders the world map and D3 Scale supports mathematical mappings from event values to visual encodings. The interface includes continental classification, geographic markers, active-zone highlighting, directional indicators where applicable and event popups.

3.5 Presentation and User Layer

Tailwind CSS styles the interface, Framer Motion supplies interface animation, and Lucide React supplies icons. Responsive layouts support both desktop and mobile interaction. Users can inspect the map, navigate highlighted zones, view the chronological event feed, open event details and follow source links.

IV. DATA PROCESSING METHODOLOGY

The processing pipeline consists of the following stages:

1. Data Acquisition: Relevant event logs are retrieved from GDELT V2 endpoints through the application's API/data handling layer.
2. Contextualization: Selected records are cross-referenced with contextual information described by the project, including known conflict or casualty context where available.
3. Parsing: Application-level parsing/NLP heuristics extract or normalize actor names, geographic locations and event types.
4. Event Object Formation: Processed information is converted into a consistent event structure containing fields such as event ID, source, event type, location, actors, intensity and timestamp.
5. Visualization: Geographic data are rendered using React Simple Maps, while D3 Scale supports visual encodings for map and interface elements.
6. Frontend Integration: Processed event objects are connected to reusable React components within the Next.js App Router architecture.
7. Responsive Design: Layouts and controls are designed to remain usable across desktop and mobile displays.
8. Testing and Evaluation: The system is intended to be tested for data loading, map rendering, event display, source-link behavior, responsiveness and API/data constraints.

The methodology is designed to avoid presenting raw source data directly. Instead, the project creates an application-level representation that links geographic position, event category, contextual information and source provenance.

V. IMPLEMENTATION

The frontend is implemented as a Next.js and React application using the App Router. TypeScript is used to maintain a structured and statically typed codebase. Tailwind CSS provides utility-based styling, while Framer Motion and Lucide React support interface animation and iconography. React Simple Maps is used for SVG-based world-map rendering and D3 Scale supports data-to-visual mappings.

Axios is used for HTTP communication with the data/API layer. The project retrieves information from GDELT V2 endpoints and prepares event objects for presentation. The documented repository structure places Next.js pages and API logic under src/app, reusable interface elements under src/components and supporting utilities/API clients under src/lib.

The principal functional modules are:

- Data Acquisition Module: Retrieves raw event logs from GDELT V2 endpoints.
- Contextualization Module: Cross-references events with known conflict/casualty context.
- Parsing Module: Extracts actors, locations and event types using parsing heuristics.
- Event Object Module: Converts processed information into a consistent application-level event structure.
- Interactive Map Module: Renders geographic visualization using React Simple Maps and D3 Scale.
- Regional Classification Module: Applies continental color coding for geographic orientation.
- Active Zone Highlighting Module: Identifies and highlights high-activity regions.
- Event Popup Module: Displays contextual details, source links and intensity information.
- Conflict Feed Module: Presents a chronological list of processed events.
- Responsive Interface Module: Provides collapsible navigation and touch-friendly controls.

The current project synopsis does not document a separate relational database or a user-authentication subsystem; therefore, these are not treated as implemented modules in this paper.

VI. EVALUATION PROTOCOL AND EXPECTED OBSERVATIONS

6.1 Evaluation Areas

The project synopsis specifies testing of data loading, map rendering, event display, source-link behaviour, responsiveness and handling of API/data constraints. These areas can be used to assess whether the dashboard performs its intended visualization and interaction functions.

6.2 Functional Evaluation

Functional evaluation should verify that event information can be acquired from the documented source, processed into event objects and displayed as map makers and feed items. Event popups should expose available location, actor, intensity and source information, while source links should allow users to inspect the underlying reporting.

6.3 Usability and Responsiveness

The responsive interface should be checked on desktop and mobile displays. Navigation, map controls, event popups and the collapsible navigation/drawer should remain usable on smaller screens. The evaluation should also consider whether the combination of map and chronological feed makes geographic and temporal relationships easier to explore.

6.4 Data Quality and Traceability

Because GDELT is derived from news/media reporting, evaluation should account for incomplete, duplicated, delayed or incorrectly interpreted records. Source links are important for traceability and should remain available wherever the underlying data provides them.

6.5 Results Status

The supplied synopsis describes the evaluation scope but does not provide measured accuracy, latency, usability scores or a completed experimental results table. Therefore, this review does not invent numerical results. Such values should be added after actual testing of the implemented system.

VII. DISCUSSION AND LIMITATIONS

WORLD CONFLICT MAP provides a unified visualization layer for distributed geopolitical event information and reduces the effort required to connect events with locations. Combining an interactive map with a chronological feed gives users both spatial and temporal perspectives. Direct source links improve traceability, while the component-based architecture makes the application extensible.

However, the system has important limitations. GDELT is derived from news/media reporting and therefore reflects source coverage and extraction limitations. Event records can contain ambiguity, duplication, missing information or incorrect geographic and actor interpretation. The application's parsing heuristics cannot guarantee semantic accuracy for every event.

Contextual casualty information may be incomplete or historical and should not be interpreted as a live authoritative count. API availability, rate limits and network connectivity can affect freshness. The visualization also simplifies complex geopolitical situations and may omit important context. Source links provide traceability but do not themselves establish that every claim is independently verified.

The application should therefore be treated as an information-aggregation and visualization tool for educational and research-oriented use, not as a substitute for professional verification, official government information or primary reporting.

VIII. CONCLUSION AND FUTURE WORK

This paper presented WORLD CONFLICT MAP, a web-based real-time/frequent-update global conflict monitoring dashboard that integrates GDELT event information with interactive

geographic visualization and a chronological event feed. The system follows a Fetch, Contextualize, Parse, Display workflow and uses Next.js, React, TypeScript, Tailwind CSS, React Simple Maps, D3 Scale and Axios to transform event information into an accessible visual interface.

The project demonstrates how public event data, API integration, application-level processing and geospatial visualization can be combined to support exploration of global geopolitical events. Its design emphasizes geographic context, event-level details, source traceability and responsive interaction while explicitly recognizing the limitations of news-derived event data.

Future work includes adding multiple independent event/conflict datasets, stronger event deduplication and entity resolution, formal confidence/provenance indicators, historical time-series analysis and playback, advanced filtering by country, region, category, actor and time, server-side caching and queue-based processing, and a persistent database for historical storage where appropriate. Additional directions include user accounts and configurable alerts, improved NLP-based event extraction, automated data-quality checks, accessibility and multilingual support, broader geopolitical event analysis and export functionality for research datasets and reports.

REFERENCES

[1] The GDELT Project. Official documentation and event database: GDELT 2.0 and the CAMEO taxonomy.
[2] The GDELT Project. GDELT 2.0 data architecture: Event and mention documentation.
[3] Vercel. Next.js documentation: App Router and Server Components.
[4] Meta Open Source. React documentation.
[5] Microsoft. TypeScript documentation.
[6] Tailwind Labs. Tailwind CSS documentation.
[7] React Simple Maps. Documentation: Composable Map, Geographies and geographic projections.
[8] D3.js. D3 Scale documentation.
[9] Framer Motion. Framer Motion documentation.