Autonomous Trade Conflict Simulation

Documentation and Analysis Report

Project: Trade War Multi-Agent System

1. Executive Summary

This project implements an autonomous multi-agent simulation to model trade conflict dynamics among three countries. The system continuously updates tariffs, trade flow, strategic posture, and welfare through interacting agents (country strategy, market intelligence, game theory, negotiation, and scenario control). A live dashboard visualizes policy and welfare evolution in real time.

2. Problem Definition

The objective is to simulate how trade conflicts evolve under repeated strategic interaction, retaliation, and negotiation pressure.

The system is designed to:

- *Model tariff escalation and rollback decisions.*
- *Estimate welfare and trade-flow impact over time.*
- *Detect windows for negotiation.*
- *Provide transparent, event-level reasoning traces.*

3. Model Overview

Model class:

- *Multi-agent, event-driven simulation*
- *Rule-based economic transitions with optional LLM-assisted reasoning*
- *Continuous state publishing for dashboard visualization*

Entities:

- Countries: Country Alpha, Country Beta, Country Gamma
- Sectors: electronics, automotive, agriculture, energy, manufacturing
- Main policy instrument: tariff updates (especially electronics)

4. Architecture

Execution flow:

1. `backend/run.py` starts HTTP server and simulation runtime.
2. `backend/system.py` initializes world state and all agents.
3. Agents communicate via `EventBus`.
4. `state_publisher` updates trade flow and welfare every cycle.
5. Snapshots are written to `shared/state.json` and `shared/events.json`.
6. Dashboard polls and renders live indicators.

Core modules:

```
`backend/agents.py`
`backend/world.py`
`backend/tools.py`
`backend/scenario_agent.py`
`backend/llm.py`
`backend/storage.py`
`dashboard/app.js`
```

5. Agent Design

Country Strategy Agents (Alpha, Beta, Gamma)

Use trade position, tariff impact, and retaliation risk tools.

Update tariff rates and emit policy events.

Alpha includes learning feedback:

- welfare delta tracking
- aggression adaptation
- effectiveness score update

Market Intelligence Agent

Monitors flow and policy changes.

Publishes sector exposure and impact memos.

Game Theory and Strategy Agent

Produces strategic classification and equilibrium-oriented guidance.

Negotiation and Resolution Agent

Detects negotiation windows.

Proposes phased rollback plans when conditions are favorable.

Scenario Execution Agent

Phase cycle:

1. Cold Start
2. Initial Shock
3. Escalation
4. Strategic Adaptation
5. Negotiation or Collapse
6. Learning Update

6. State and Dynamics

Country state includes:

GDP

Sector-wise exports/imports

Sector-wise tariffs

Political pressure

Strategic posture

Welfare

Simulation cycle includes:

Gravity-style trade flow update

Sector drift

Welfare recomputation

Agent reasoning/action publication

7. Inputs and Outputs

```
Inputs:
- Initial country and sector parameters
- Recent event memory per agent
- Runtime configuration: `LLM_PROVIDER`, API keys, `REDIS_URL`

Outputs:
- `shared/state.json` (latest system snapshot)
- `shared/events.json` (recent event list)

Key fields for assessment:
- `phase`, `round`, `classification`
- `countries[*].tariffs`
- `welfare_impact`
- `learning.alpha_aggression`
- `learning.effectiveness_score`
- `policy_timeline`
```

8. Analysis

8.1 Evaluation Questions

Does escalation improve strategic outcomes or reduce welfare?

When does retaliation risk force policy rollback?

How often do negotiation windows occur?

Does Alpha's learning improve outcomes over rounds?

Based on the implemented agent rules and observed runtime behavior:

***Escalation vs welfare**: Escalation can improve short-term strategic pressure, but repeated escalation generally lowers welfare over time due to tariff drag and trade-flow disruption.*

***Rollback trigger conditions:** Policy rollback becomes more likely when retaliation risk is high (risk-aware strategy path) and welfare trends deteriorate.*

***Negotiation frequency:** Negotiation windows occur intermittently rather than continuously, typically after sustained pressure/loss conditions are detected.*

***Alpha learning effect:** Country Alpha shows adaptive behavior through aggression and effectiveness-score updates. Early rounds indicate policy adaptation, but longer simulation runs are required to confirm stable improvement.*

8.2 Quantitative Metrics

Welfare trend per country (round-wise)

Average tariff by country/sector

Escalation-to-rollback action ratio

Negotiation-window frequency

Alpha effectiveness score trend

8.3 Expected Behavioral Patterns

High retaliation risk should increase rollback actions.

Low risk plus aggressive stance should increase escalation frequency.

Welfare degradation should pressure de-escalation over time.

Learning in Alpha should adjust aggression to stabilize welfare.

8.4 Interpretation Guidance

Use model outputs as scenario insights under simplified assumptions. Treat findings as directional, not predictive of exact real-world policy outcomes.

9. Robustness and Reliability

LLM reasoning timeout guard in agent loop.

Automatic fallback to heuristic mode when API or timeout constraints occur.

Structured event outputs improve traceability and debugging.

10. Limitations

Limited policy instrument set (tariffs dominate).

No historical data calibration yet.

Single-process runtime with in-memory event bus.

No full automated test suite.

Redis mode is not directly mirrored to file-based dashboard path by default.

11. Ethical Note

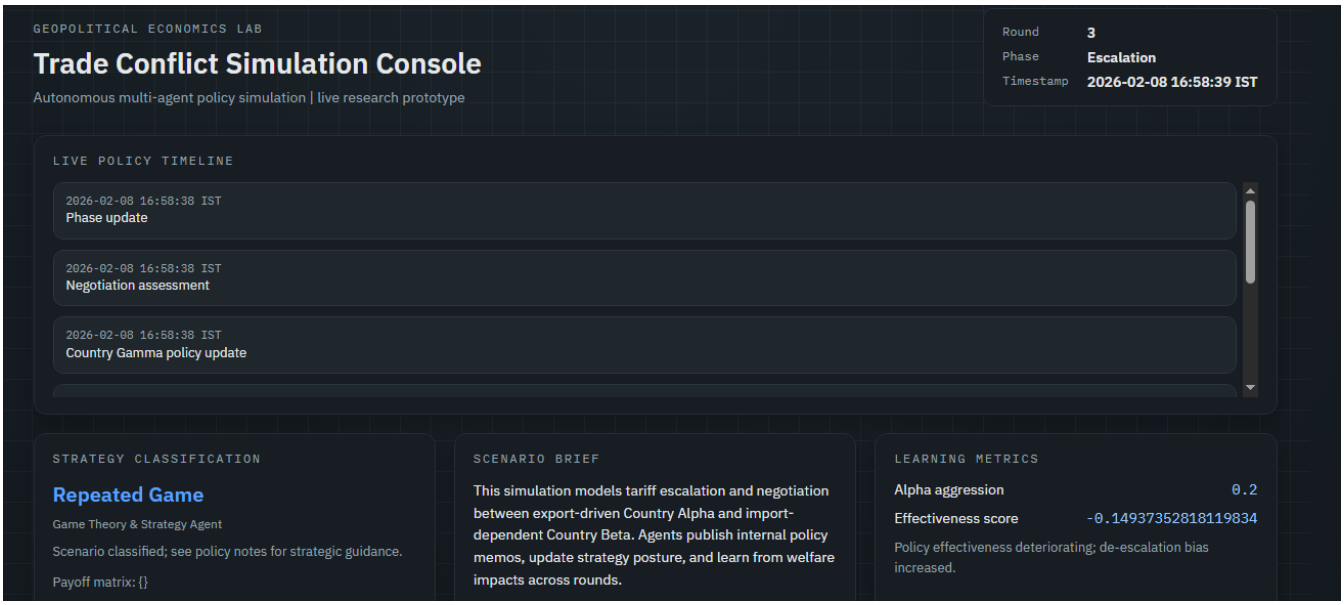
This simulation is for educational and exploratory policy analysis. It is not a production policy recommendation engine.

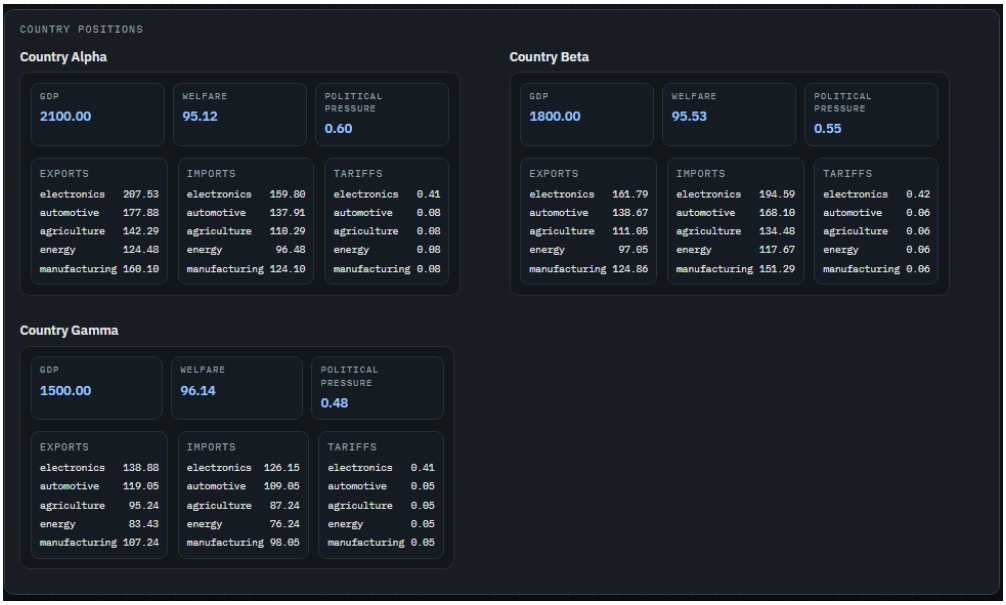
12. Reproducibility

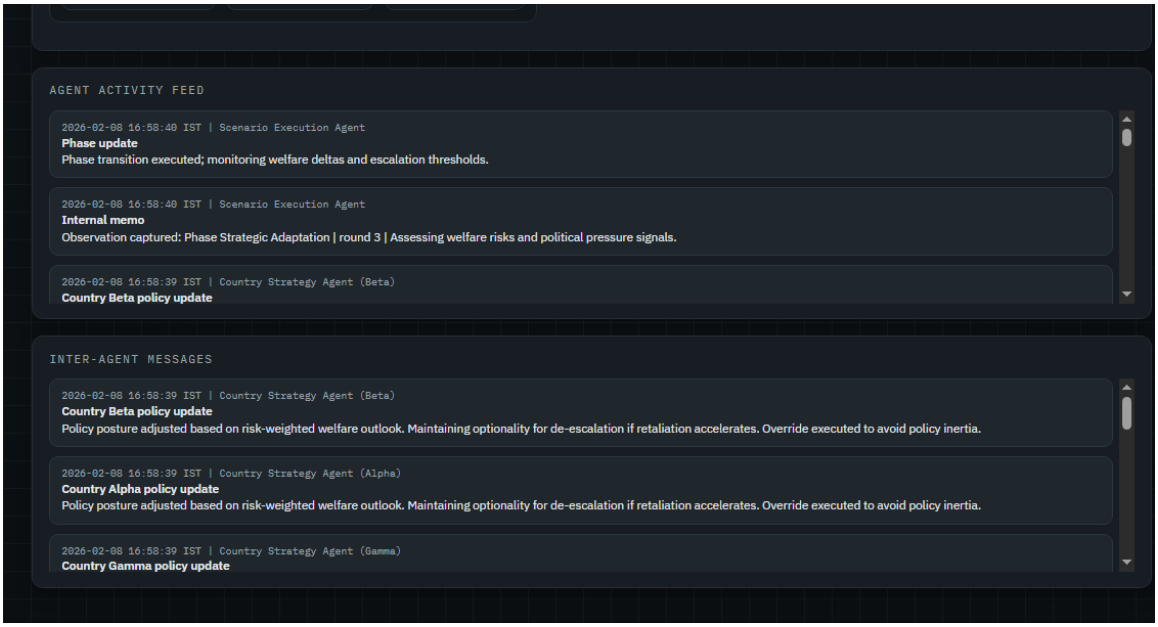
```powershell
pip install -r requirements.txt
python backend\run.py
```

Open: `http://127.0.0.1:8000`

13. Snapshots







The dashboard provides real-time visibility into tariff changes, welfare trends, and negotiation signals.

Policy timelines allow tracing agent decisions to specific events.

Live state polling enables interactive scenario observation without pausing simulation execution.