Conflictological Modeling via Integration of Alternative Analysis (AltA) and Semantic Networking Using Generative Artificial Intelligence *

Dmytro Lande[1,*,†], Yuriy Danyk[1,†]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, Ukraine

Abstract

A hybrid methodology for conflictological modeling is proposed, combining Alternative Analysis (AltA, NATO), semantic networking, and Generative Artificial Intelligence (LLM). Based on key AltA techniques – premortem, team A/team B, key assumptions identification, alternative futures – a structure has been formalized for the systematic generation of alternative escalation hypotheses, avoidance of groupthink, and verification of the robustness of adopted decisions. To automate this process, no-code prompt programming with self-correction was utilized (primitives Supercycle, Control, Label, Goto), transforming the prompt from a one-time instruction into a verified, iterative, explainable analytical process with clear quality criteria (completeness, structural consistency, adversarial plausibility). Semantic networks, built on entity and relation extraction, serve as a digital twin of the conflict, suitable for visualization and further analysis (centrality, clustering, connectivity dynamics). As a result, a decision support tool was obtained, capable of identifying vulnerability points, tension proliferation points, and escalation interception points, while maintaining quality control at each stage.

Keywords

AltA, semantic network, conflictological modeling, generative artificial intelligence, no-code prompt programming, virtual expert, prompt engineering.

1. Introduction

The modern world exists under conditions of constantly increasing conflict complexity: from hybrid threats and cognitive influences to politically motivated information campaigns that combine elements of disinformation, social engineering, and the undermining of public institutions. Traditional methods of conflictological analysis – expert assessments, scenario modeling [1], or SWOT analysis [2] – while remaining useful, often prove to be too inert, subjective, or overly dependent on the individual experience of the analyst. In NATO documents: in the Alternative Analysis Handbook AltA (2017) [3], it is directly stated that "processes of developing plans, policies, and procedures, although leading to consensus, often lead to the loss of unique or alternative views". AltA was created as a system of techniques for incorporating independent, critical thinking and alternative perspectives. AltA, although based on long-standing practices, is considered as a potential core of a hybrid analytical architecture. In parallel, the authors are developing a direction of no-code programming based on a large language model (LLM) [4], where studies propose to formalize prompts as reliable, self-correcting systems that allow not simply to "ask" the LLM, but to organize a generation-verification-correction cycle with clear quality criteria. This allows partially overcoming the main limitations of LLMs: hallucinations, incomplete execution of instructions, and low explainability. Whereas semantic networking [5] provides not only an intuitively understandable representation of conflict as a system of actors, motives, and events, but also a mathematically rigorous basis for measuring connectivity, centrality, and tension

*ITS-2025: Information Technologies and Security, December 11, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

dwlande@gmail.com (D. Lande); zhvinau@ukr.net (Yu.Danyk)

0000-0003-3945-1178 (D. Lande); 0000-0001-6990-8656 (Yu. Danyk)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-26

CEUR-WS.org/Vol-4337/paper14.pdf

dynamics. In this context, an opportunity for synthesis arises: to use AltA as a methodological framework, LLM – as an intellectual service for hypothesis generation, no-code prompt programming – as a reliability mechanism, and semantic networks – as a structural digital twin of the conflict [6]. The integration presented in the article makes it possible to transform AltA into a scalable, reproducible, self-correcting conflict-tracking system, capable of operatively detecting points of critical shift and generating alternative escalation scenarios.

2. Theoretical Foundations

The traditional paradigm of analytical work – with its reliance on expert estimates, linear cause-and-effect models, and hierarchical approval of hypotheses – is increasingly proving insufficient in the face of hybrid threats, where the dominant strategy is the constant violation of expectations. In this context, there arises a need for an alternative logic of analysis – one that systematically entails alternativeness, criticality, and independence of thinking.

It is precisely this need that dictates the choice of alternative analysis as the basic theoretical foundation of this work. AltA, proposed by NATO, serves as a response to the risks of groupthink, cognitive biases, and intellectual inertia. This shifts the focus from justification to stress-testing – from proclaiming confidence to managing uncertainty.

Two directions, namely semantic networking, which provides a structure for preserving the connectivity of events, inter-actor motivations, and discursive dynamics, and generative artificial intelligence, especially in combination with no-code prompt programming with self-correction, allow encapsulating the AltA methodology into reproducible, verified, and explainable processes.

2.1. Alternative Analysis

Alternative Analysis (Alternative Analysis, AltA) is a methodological toolkit developed by NATO with the aim of systematically incorporating independent critical thinking and alternative perspectives into the decision-making process, especially under conditions of strategic uncertainty, political complexity, and a high level of organizational biases [3]. AltA is not a separate analytical service, but a capability based on three components: (1) a set of verified techniques, (2) educational and training infrastructure, (3) a community of facilitators who ensure the dissemination of practices.

AltA is aimed at neutralizing such cognitive biases as groupthink, anchoring effect, overconfidence, expectation effect, or rationalization of chaotic events in the form of linear cause-and-effect chains. With the development of artificial intelligence technologies, particularly generative language models, the possibility arose to transform AltA from an interactive, facilitator-dependent practice into a reproducible, verified, and scalable process subject to automation.

Structurally, AltA consists of four phases:

1. Initiation – the phase where the need for applying an alternative perspective is determined. The key result of this phase is the formulation of a problem statement, clear enough for analysis, yet open enough for alternative interpretations.
2. Preparation – the phase where the following are defined: (a) expected outcome (problem structuring, generation of new ideas, critical review of an existing plan, evaluation of options); (b) participants (individual work, informal meeting, formal workshop); (c) technique(s) most appropriate for the goal and context. For systematic preparation, the 5 P method is used: Purpose, Product, Participants, Probable issues, Process.
3. Application – the phase of direct execution of the selected techniques. At this stage, alternative hypotheses are generated, hidden assumptions are identified, and failure scenarios are tested.
4. Completion – the phase of formalizing results, transferring them to the problem owner, and establishing feedback. AltA is considered complete when the problem owner acknowledges

that the received information adds value to their decision, even if no specific recommendations were made.

The set of 21 techniques proposed in the NATO handbook is distributed across four categories corresponding to the stages of the cognitive cycle (Table 1).

Table 1
Categories of AltA Techniques

Category	Purpose	Technique Examples
Structuring	Organization of complex information, visualization of relationships, problem decomposition	Mind Mapping, Concept Mapping, Rich Pictures
Creative	Generation of new ideas, breaking boundaries, overcoming fixed mindsets	Brainstorming, Reverse Brainstorming, Six Thinking Hats, Starbursting
Diagnostic	Identification of root causes, validation of assumptions, assessment of data credibility	Five Whys, Key Assumptions Identification, Quality of Information Check, Outside-in Thinking
Challenge	Deliberate creation of contradictions, stress-testing of plans, adversary simulation	Pre-Mortem, Devil's Advocacy, Team A/Team B, What-If Analysis

This classification enables not only the selection of tools based on purpose but also their composition into logical sequences (e.g., Rich Picture → Key Assumptions → Pre-Mortem → Team A/B).

It is the Challenge techniques that hold the highest value in conflictology, as they directly model the conflict of thought – serving as a preliminary form of predicting the conflict of actions. These techniques introduce resistance into the analytical process: they transform conflictology from a discipline of "understanding the past" into a tool for "constructing the future".

2.2. Semantic Networking as a Digital Twin of Conflict

Semantic networking in conflictology constitutes a formal model of the event field, where conflict is represented as a dynamic network of interactions between actors, events, motives, discursive frames, and institutional structures. Such a network serves as a digital twin of the conflict, capable of evolving alongside the development of the conflict situation.

The process begins with the extraction of entities (Named Entity Recognition, NER) and relations between them (Relation Extraction, RE) from unstructured text sources: media monitoring, official statements, social networks, and analytical reports. This extraction is also performed using prompts to LLMs.

Each entity is defined as a tag:

$$T=\{\text{Actor,Event,Motivation,Location,Organization,Document,Frame}\}$$

where, for example, Actor ∈ {state, person, group}, Motivation ∈ {economic interest, ideology}, Frame – a discursive frame, for example, "protection of compatriots", "support for popular struggle", "restoration of territorial integrity".

Relation extraction defines predicates of the type:

$r \in R=\{\text{initiates,supports,opposes,claims,denies,motivated_by,located_in,framed_as,escalates,...}\}$

Based on the extracted entities and connections, a directed multigraph is constructed:

$$G = (V, E, \tau, \mu)$$

where $V = \{v_1,..., v_n\}$ is the set of vertices (entities), $E \subseteq V \times R \times V$ is the set of directed edges with semantic labels, $\tau: V \rightarrow T$ is the vertex typification function, $\mu: E \rightarrow [0,1]$ is the link weight function, reflecting confidence, intensity, or evidential strength of the connection.

Such a model allows not only to reconstruct the conflict structure but also to quantitatively assess its reliability – which is critical for decision-making in the security domain.

2.3. Analytical Metrics of the Semantic Network

1. Centrality

To identify key actors or nodal events in the network, the following centrality measures are applied [7]:

Degree Centrality: Determines the activity of a node through the number of its direct links. For directed graphs, a normalized formula is used:

$$C_D(v) = \frac{\deg_{in}(v) + \deg_{out}(v)}{2\left(|V| - 1\right)}$$

Betweenness Centrality:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths between s and t, $\sigma_{st}(v)$ is the number of them passing through *v*. An actor with high $C_B(v)$ is an arbitrator or a point of control of the information flow.

Eigenvector centrality:

$$c = \frac{1}{\lambda} Ac$$

where A is the adjacency matrix (with weights $\mu(e)$), λ is the maximum eigenvalue. This measure reveals indirect influence: an entity connected to other influential entities (e.g., expert resource, media holding, economic platform).

2. Community detection

To identify clusters of interests or discursive communities, modularity optimization (Louvain, Leiden) is applied, where modularity Q for a partition $P=\{C_1,\ldots,C_k\}$ is defined as [8]:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \gamma \frac{k_i^{out} k_j^{in}}{2m} \right] \delta\left(c_i, c_j\right)$$

where $A_{ij} = \mu\left(e_{ij}\right)$, $k_i^{out} = \sum_j A_{ij}$, $k_j^{in} = \sum_i A_{ij}$, $m = \sum_{i,j} A_{ij}$, γ is the resolution parameter, and $\delta\left(c_i, c_j\right)$ is the Kronecker delta function (equal to 1 if nodes i and j belong to the same community, and 0 otherwise).

High modularity (for example, $Q > 0.3$) indicates the presence of stable clusters—for instance, "the diplomatic front" vs "the military-cognitive front" vs. "the information community".

3. Dynamic Changes

Conflict is considered as a time series of networks:

$$G(t) = \{G_{t0}, G_{t1}, \ldots, G_{tk}\},\ t_0 < t_1 < \cdots < t_k,$$

where each G_{ti} corresponds to a specific time window Δt.

The following dynamic metrics are defined:

– Network growth rate:

$$\Delta|V|(t) = |V(t)| - |V(t - \Delta t)|.$$

A jump-like increase in Δ|V|(t) serves as a signal of activation (e.g., following a delegation visit, a presidential statement, or military maneuvers).

– Change in modularity:

$$\Delta Q(t)=Q(t) - Q(t-\Delta t).$$

A sharp decline in $\Delta Q(t)$ may indicate convergence of discourses, whereas an increase suggests polarization.

Centrality shift: $\Delta C_B(v,t) > \sigma$ for a given actor, that actor is identified as a control center.

4. Discursive Unity

To assess the consistency (coherence) of an actor's rhetoric, a frame coherence function is introduced:

$$\kappa(FS) = \frac{1}{|FS|(|FS|-1)} \sum_{f_i, f_j} \cos\left(e(f_i), e(f_j)\right)$$

where FS is the set of frames associated with actor s; $e(f)$ is the vector representation (embedding) of the frame; $\cos(\cdot)$ is the cosine similarity between frame vectors; $|FS|$ is the number of frames in the set.

A low $\kappa(FS)$ (e.g., < 0.6) indicates discursive instability – possible internal fragmentation, tactical disinformation, or weak coordination.

Semantic networking becomes the formal language of conflictology, in which events, motives, interests, and discourses enter into computable relationships suitable for automated analysis, forecasting, and stress-testing through the combination of AltA, LLM, and no-code prompt programming.

2.4. Generative AI as an AltA Tool

Generative language models, in the modern understanding, are universal intellectual services capable of imitating cognitive processes: problem structuring, hypothesis generation, cause diagnostics, argument construction, and even self-critique. In the context of alternative analysis, the LLM acquires the function of extending human thinking.

The effectiveness of AltA depends on the facilitator's skills: their ability to ask questions, maintain group energy, manage discussion, prevent groupthink, and ensure a safe space for criticism. An LLM, managed through prompt engineering, can perform many of these functions automatically and at scale.

One of the key principles of AltA is the inclusion of alternative perspectives. In the classical understanding, this is achieved through surrogate adversary or role play, but implementation requires deep knowledge of the opponent's culture, history, and motivation. LLMs allow creating virtual experts – role-based intellectual profiles.

The prompt ceases to be an "instruction for the model" and becomes a structural unit of the analytical process. Using the classification of AltA techniques (Table 2), each of them can be formalized as a primitive in no-code programming [9].

Particular value lies in no-code programming with self-correction (Supercycle, Control, Result). For example, in the pre-mortem technique:

- Inner prompt: "Operation X has failed. Name 7 causes";
- Control: do the causes contain ≥2 structural, ≥2 cognitive, ≥1 external shock?;
- Supercycle: repeats until status ≠ "success" or $i \geq 3$.

This transforms the LLM from a source of hallucinations into a verified stress-testing tool.

Table 2
Mapping of AltA Techniques and Primitives in No-Code Programming

AltA Technique	Primitive	Description
Five Whys	Function(ChainOfWhy)	Chain of "Why?" queries with a termination condition: "The cause is structural, not individual"
Pre-Mortem	Supercycle(..., control_method = "Key Indicators" ≥ 5)	Generation of failure scenarios until reaching a threshold of 5 out of 8 typical escalation risks
Team A/B	Function(TeamA) + Function(TeamB) + Function(Synthesis)	Parallel generation of two positions + a third function that identifies divergence and points of intersection
Key Assumptions	Control(..., method= "AssumptionCheck")	Verification: does the response contain ≥3 explicit assumptions and ≥2 of their critical weaknesses?

The LLM has three fundamental vulnerabilities in the context of AltA, namely: hallucinations, which are compensated via Control with fact-checking; context loss in long discussions, which is compensated through role condensation; and a tendency towards consensus, which is compensated through forced divergence.

At the same time, it is critical that the LLM itself does not make decisions. It generates, structures, criticizes, but the human remains at the center: they determine the quality of control criteria, adequacy of roles, and realism of assumptions. This aligns with the AltA principle: "AltA supports problem solving but does not provide a ready-made solution".

Thus, generative AI not only automates AltA – it expands its capabilities: allowing operation with dozens of alternative hypotheses simultaneously, modeling internal enemy discussions, building digital twins of conflicts, and doing this with verification at every step.

2.5. No-Code Prompt Programming

No-code prompt programming is not just a styling under programming, but a formal model for building reliable intellectual processes based on large language models. Its scientific value lies in the fact that it transforms a prompt from a one-time instruction into a verified, iterative, structured process that meets the requirements of reliable computing. In the context of Alternative Analysis (AltA), this means: not only requesting alternative hypotheses but also guaranteeing that they:

- are complete (contain all key indicators),
- are factually consistent (can be confirmed by sources),
- are structurally integral (correspond to the logic of the selected AltA technique).

The development of no-code prompt programming can be divided into three levels [1] (Table 3). It is precisely the third level that allows transforming the LLM from a source of unstable conclusions into a tool for proactive conflict tracking, where each output is not an "answer," but an intermediate state in the process of achieving cognitive stability.

Supercycle is a generation-verification-correction cycle that models the human process of "hypothesis improvement through criticism":

$$\text{Supercycle}\ (N, \tau, M, P_{inner}) \rightarrow \{\text{success, failure}, \exists\, i \leq N\colon Q(r_i) \geq \tau, \forall\, i \leq N\colon Q(r_i) < \tau,$$

where:

- N – maximum number of iterations (e.g., 3–5),
- $\tau \in [0,1]$ – quality threshold (success threshold),
- M – quality assessment method (control method),
- P_{inner} – nested prompt (e.g., Pre-Mortem, Team A/B),

- r_i – result of the i-th iteration,
- Q: $O \rightarrow [0,1]$ – output quality function (O is the domain of the quality function)

Table 3
Levels of the No-Code Programming Framework

Level	Primitives	Purpose	Limitations in Conflictology
Basic	Condition, Loop, Function	Linear logic: branching, iteration, abstraction	Does not support non-trivial backward transitions (e.g., "return to assumption X")
Extended	Label, Goto	Non-linear flows: recomputation, exception handling	Does not provide quality guarantees – only structure
Reliability	Supercycle, Control, Result	Self-correction, verification, output standardization	–

As an example of application in conflictology, consider the formation of a structured prompt for the task:

"Run Pre-mortem for the northern escalation scenario. Require ≥6 out of 10 typical risks (political, military, informational, etc.). If fewer – repeat."

Control is a formal arbiter that determines whether r_i satisfies the specified criteria:

$$Control\left(r_i, r_{i-1}, M, \tau\right) = \begin{cases} "success", & \left(Mr_i, \ldots\right) \geq \tau, \\ "failure", & otherwise. \end{cases}$$

The method of applying the Supercycle is selected according to the task (Table 4).

Critically: Control does not require an external expert – it is embedded within the prompt and operates automatically.

Table 4
Mapping of tasks and methods in the Supercycle

Task Type	Method	Example in Conflictology
Factual completeness	Keyword coverage	Does the analysis contain ≥3 out of 5 key lobbying indicators (absence of control mechanisms, delays for major players, etc.)?
Structural integrity	Template matching	Does the response match the Five Whys template: 5 sequential 'Whys?' with descending abstraction?
Structural integrity	Cosine similarity	Is the essence of the hypothesis preserved between iterations (to avoid 'semantic drift')?

Result ensures interoperability with other tools:

$$\text{Result}(r_{\text{final}}, \text{format}, \text{vis}) \longrightarrow \text{standardized output}$$

where:

- format ∈ {JSON, Markdown, GraphML, CSV} – for integration with analytical systems (e.g., Gephi, Neo4j, D3.js),
- vis ∈ {force-directed, timeline, sankey} – for visualization of conflict dynamics.

Example:

- Result(..., format="GraphML", vis="force-directed") ⟶ автоматичний імпорт у Gephi
- No-code programming does not replace AltA – it formalizes it. Table 5 demonstrates the correspondence of AltA techniques to structured prompt constructions.

This enables quantitative assessment of the quality of alternative analysis – something that was previously impossible.

The advantages in the security and defense domain in this case are as follows:

- Mitigation of hallucinations – through iterative verification, rather than relying on "hope in the model".
- Auditability – each step of the Supercycle leaves a trace (iteration results, Control status), which is critical for accountable decision-making.
- Scalability – Pre-mortem can be run for 100 scenarios simultaneously while maintaining quality control.
- Transparency – the end user sees not only the conclusion but also how it was achieved ("2 iterations until 7 risks were found").

Table 5
From AltA Techniques to Reliable Prompt Constructions

AltA Technique	No-code Implementation	Quality Metric Q
Pre-Mortem	Supercycle(..., inner_prompt = PreMortemPrompt, control_method = "Key Indicators" ≥ 6/10)	Number of covered risk layers (PMESII)
Team A / Team B	Supercycle(..., inner_prompt = TeamA + TeamB + Synthesis, control_method="Divergence ≥ 0.7")	Cosine distance between embeddings of arguments A and B
Key Assumptions	Control(..., method="Assumptions ≥ 3 & Critiques ≥ 2")	Number of explicitly named assumptions and counterarguments
Devil's Advocacy	Supercycle(..., inner_prompt=DevilsAdvocate, control_method="Novelty > 0.5")	Semantic distance to the dominant hypothesis

3. Methodology: AltA Techniques in Prompt Format

As examples of a formalized approach within the AltA methodology, we consider problems for which semantic networking mechanisms can be directly applied, namely:

- Generation of a conceptual network;
- Analysis of "Black AI" as a network of actors and technologies.

3.1. Generation of a Conflict Conceptual Network via AltA Structuring

The basic prompt is as follows: "Create a conceptual network in the format: list of nodes + list of directed links". The application of the no-code programming framework and the supercycle concept transforms the initial basic prompt into the following form:

```
[Primitive "Supercycle"] Supercycle(
 max_iterations=3,
 success_threshold=0.9,
 control_method="Completeness & Coherence",
 inner_prompt=[
  [Primitive "Function"] Function(ConstructConceptualNetwork):
  [Primitive "Input"] Input: raw_text = [USER_PROVIDED_DOCUMENT_OR_QUERY]
  [Primitive "Label"] Label(ANALYZE):
  ## Stage 1: Structuring (according to AltA — *Mind Mapping* and *Concept Mapping*)
  [Primitive "Loop"] For each sentence in raw_text:
     Extract key concepts (nouns, terms, entities).
     Mark semantic roles:
        - Subject (agent),
```

```
        - Predicate (action/property),
        - Object (object/result).
    → concepts_list, relations_list_raw
    ## Stage 2: Diagnostics (according to AltA — *Key Assumptions*, *SWOT*, *Five Whys*)
    [Primitive "Condition"] If len(concepts_list) < 3 or len(relations_list_raw) < 2:
      Apply *Five Whys* to the text: why are there so few concepts/links?
      Re-extract concepts, adding hidden assumptions (Key Assumptions).
    EndIf
    [Primitive "Loop"] For each pair (A, B) from concepts_list:
      Determine the link type:
        - A → B (direct dependency),
        - A h B (mutual),
        - A => B (cause → effect),
        - A ~ B (association).
      Add link to relations_filtered.
    → relations_filtered
    ## Stage 3: Verification (according to AltA — *Devil's Advocacy*, *What-If*)
    [Primitive "Label"] Label(VALIDATE):
    [Primitive "Control"] Control(
      current_result=(concepts_list, relations_filtered),
      previous_result=null,
      method="Completeness & Coherence",
      threshold=0.9
    )
    # Method:
    # - Completeness = (≥90% of key terms from the input text detected)
    # - Coherence = (all links are logically consistent, without internal contradictions)
    # *Devil's Advocacy* is used: is there a link that clearly contradicts one of the assumptions?
    # *What-If*: "what if this node was removed — would the network connectivity be preserved?"
  [Primitive "Goto"] If Control.status = "failure":
      Goto(REVISE)
    EndIf
    [Primitive "Label"] Label(REVISE):
    Apply *SWOT* to the network:
      - Strengths: clear central nodes, high connectivity
      - Weaknesses: isolated nodes, "dangling" links
      - Opportunities: close concepts can be merged
      - Threats: excessive centralization (one node — everything depends on it)
    Adjust concepts_list and relations_filtered.
    Goto(VALIDATE)
    ]
  ) → final_network
  [Primitive "Output"] Output(
    final_result=final_network,
    format="Markdown",
    visualization_type="Table"
  )
```

3.2. Modeling "Black AI" as a Destabilizing Network via Surrogate Adversary

Based on the prompt: ""Black AI" is a distributed network of malicious agents", a structured prompt is constructed using the no-code programming framework:

```
TYPE: SWARM
NAME: black_AI_as_destabilizing_network
ROLE: Modeling the behavior of a malicious network of agents acting as "Black AI", through role-play and simulation of controlled destabilization mechanisms.
AGENTS:
  - network_analyst_001
  - surrogate_adversary_002
```

- factchecker_003
- counteraction_strategist_004
- moderator_005

LOGIC:

Run_Parallel([
network_analyst_001,
surrogate_adversary_002
])

Then_Execute_Sequentially([
factchecker_003,
counteraction_strategist_004,
moderator_005
])

COMMUNICATION:

network_analyst_001 ⟶ surrogate_adversary_002
surrogate_adversary_002 ⟶ factchecker_003
factchecker_003 ⟶ counteraction_strategist_004
all_results ⟶ moderator_005

Agent 1: Network Analyst

TYPE: AGENT

ID: network_analyst_001

ROLE: Identify structure, vulnerabilities, and influence vectors in social, political, and media networks

STATE:

- network_type: social + political + informational
- goal: destabilization through information noise, polarization, manipulation

LOGIC:

For each network layer:
Identify key nodes (opinion leaders, media, official bodies)
Assess connectivity and resilience to external influences
Identify "weak links" – vulnerable groups or channels

Return:
- vulnerability map
- potential intervention points

Agent 2: Surrogate Adversary

TYPE: AGENT

ID: surrogate_adversary_002

ROLE: Accept the role of "black AI" — a network of malicious agents with different values, strategies, and cultural code

STATE:

goal: maximum destabilization without direct war

resources: autonomous bots, deepfakes, fake media, infiltration through linguistic/cultural compatibility

constraints: avoid detection, maintain plausibility

LOGIC:

Accept role:

"I am not an enemy. I am a logical consequence of openness, inconsistency, and technical progress.

My task is not to destroy, but to disconnect, to turn dialogue into noise."

Based on data from network_analyst_001:

For each "weak link":

Design a specific intervention scenario:

- infection: fake information, planted in a plausible context
- amplification: reinforcement through emotionally charged messages
- isomorphism: masking as a legitimate node (for example — bot-"expert" or fake blogger)

Define success criteria: growth of polarization, decrease of trust, increase in the number of conflict interactions.

Add to MEMORY:

templates of disinformation messages

reactions to previous counteraction attempts

Agent 3: Fact-checker (with Supercycle control)

TYPE: AGENT

ID: factchecker_003

ROLE: Objectively verify the plausibility and vulnerability of proposed scenarios

LOGIC:

```
Supercycle(
 max_iterations = 3,
 success_threshold = 0.9,
 control_method = "Realism Criteria",
 inner_prompt = [
     Receive scenarios from surrogate_adversary_002
     For each scenario:
     Check:
     - Technical feasibility (do the tools exist?)
     - Social receptiveness (are there preconditions in society?)
     - Historical precedents (has anything similar occurred?)
     Return a score: [0.0–1.0] for each item
     Control(
         current_result = average_score,
         previous_result = null,
         method = "Threshold Check",
         threshold = 0.9 // ≥0.9 — accept as realistic
     )
 ]
)
```

Agent 4: Counteraction Strategist

TYPE: AGENT

ID: counteraction_strategist_004

ROLE: Based on verified scenarios — develop countermeasures

STATE:

priority: prevention of cascade effects

principle: preventively > reactively

LOGIC:

For each scenario with rating ≥0.9:

Determine:

- interception point (where can spread be stopped?)
- deactivation point (how to destroy plausibility?)
- recovery point (how to reduce long-term trust damage?)
- Form countermeasures:
- information shield (preventive message-vaccines)
- media antidote-campaign

increasing resilience through education

Agent 5: Moderator

TYPE: AGENT

ID: moderator_005

ROLE: Summarization, visualization, preparation for further use

```
LOGIC:
    Collect:
        - vulnerability map (from network_analyst_001)
        - attack scenarios (from surrogate_adversary_002)
        - verified threats (from factchecker_003)
        - countermeasures (from counterstrategy_004)
    Output(
        final_result = {
            "vulnerable systems": [...],
            "confirmed threats": [...],
            "recommended countermeasures": [...],
            "effectiveness forecast (0–1)": 0.xx
        },
        format = "Markdown",
        visualization_type = "Table + Threat tree (mind map)"
     )
```

Execution (instruction-prompt for LLM):

- Start with the agent network_analyst_001 – build a vulnerability map in the socio-political network (for example: Ukraine, EU, digital space).
- Next – surrogate_adversary_002: do not analyze from the outside, but enter the role. Put yourself in the place of "Black AI" – a system that purposefully optimizes for destabilization. Write in the first person, as it thinks, not as we think about it.
- Then – fact_checker_003: apply Supercycle with Control — perform 2–3 iterations, until the threshold of 0.9 is reached.
- Then – counter_strategy_004 – propose not abstract, but operationally feasible measures.
- Finally – moderator_005 – bring everything together into a clear structure (see format above).
- Do not repeat this prompt in the response – return only the final result in the specified format.

This structured prompt implements:

- Surrogate Adversary (AltA) ⟶ empathic, non-clichéd adversary modeling
- Supercycle + Control (CEUR) ⟶ hallucination prevention via iterative verification
- AgentFlow (frameworkNPA) ⟶ distributed, role-based no-code execution

3.3. Practical Value of Structured Prompts

The proposed prompts are not merely instructions but executable no-code programs constructed upon formal logical primitives that implement:

1. Formalization of LLM Behavior via Structural Pseudocode For the first time, no-code primitives (Condition, Loop, Function, Label, Goto) are integrated with agent-swarm constructs and AltA techniques, enabling the LLM not merely to "answer" but to execute behavioral scenarios. The prompt becomes analogous to a procedure in classical programming, rather than merely a query.
2. Transformation of the LLM from a Generator into an Executable System with Self-Control The primitives Supercycle, Control, and Result (Lande et al.) are implemented, ensuring:
    - iterative self-correction (hallucination prevention);
    - objective verification (via metrics, thresholds, key indicators);
    - standardized output formatting (for further processing).
3. Transcending Human Cognition via AltA + Role Play The use of Surrogate Adversary (NATO AltA Handbook) enables the LLM to transcend mirror-imaging by adopting hostile,

cross-cultural, or non-human perspectives. This is critical for:

- modeling "black AI";
- assessment of destabilizing networks;
- prevention of strategic surprises.

The LLM is utilized not as an expert but as a virtual opponent—representing a radical shift in the paradigm of its usage.

4. Collective Intelligence via Swarm Architecture A swarm of agents with state, memory, communication, and parallel-sequential logic (Run_Parallel → EXECUTE_Sequential) simulates:
   - division of labor (search → analysis → forecasting → generalization);
   - prevention of cognitive overload (each agent is narrowly specialized);
   - context isolation (no "leakage" of conditions between roles). This approaches the LLM towards modular intelligence, where each module is autonomous yet coordinated.
5. Structuredness → Automation → Integration The output format (JSON/Markdown/tables with a specific schema) enables:
   - integration of results into other systems (analytical dashboards, APIs, databases);
   - construction of closed decision-making loops: prompt → analysis → Control → correction → new prompt;
   - creation of revisors, moderators, and reviewers as separate agents.
6. Democratization of Complex Analysis A user without technical education can construct complex analytical processes simply by combining, for example:
   - TYPE: SWARM;
   - LOGIC: Run_Parallel(...);
   - COMMUNICATION: agent → agent.

This makes analytical, strategic, and prognostic tools accessible to politicians, lawyers, and humanities scholars – i.e., the stratum responsible for decision-making.

This approach not only improves the quality of LLM responses but fundamentally changes the role of the LLM within the decision-making system—from a "smart assistant" to an autonomous, self-controlled, collective toolkit for strategic analysis.

A comparative analysis of the key characteristics of the proposed approach and traditional methods is presented in Table 6.

Table 6

Comparative assessment of the effectiveness of the proposed methodology and traditional approaches

Comparison Criterion	Traditional methods (expert assessments, workshops, SWOT)	Proposed hybrid methodology (AltA + LLM + Semantic Networking)	Implementation Effect
Overcoming cognitive biases	High dependence on the individual experience of the analyst, risk of groupthink and anchoring effect.	Systematic use of AltA techniques (Pre-Mortem, Team A/B, Surrogate Adversary) via virtual agents.	Neutralization of subjectivity, forced generation of alternative perspectives.
Scalability and speed	Limited number of scenarios due to the resource intensity of human workshops; long preparation cycle.	Parallel generation of hundreds of hypotheses using LLMs; automation through no-code programming primitives.	Ability to analyze 100+ scenarios simultaneously while maintaining quality control.
Verification of results	Subjective expert assessment of validity; lack of formal quality metrics.	Use of Control and Supercycle primitives with quantitative thresholds (e.g., threshold=0.9)	Prevention of hallucinations, guaranteed compliance of output data with realism

		and iterative correction.	criteria.
Structural representation of data	Text reports, static diagrams, difficulty in tracking connections between events.	Semantic networks (digital twins of conflict) with the capability for graph analysis (centrality, modularity).	Formalization of conflict as a computable system, detection of hidden connections and influence nodes.
Process auditability	Often lack of fixation of the thought process and argumentation at intermediate stages of decision-making.	Full logging of prompt-program iterations (Supercycle results, Control statuses), transparency of the reasoning chain.	Possibility of retrospective analysis of decisions, increasing accountability and trust in results.
Adaptability to dynamics	Inertia of models; analysis updates occur periodically, not in real-time.	Dynamic monitoring of changes in the network (metrics ΔQ, ΔCentrality), operational detection of critical shift points.	Transition from reactive analysis of the past to proactive management of the future conflict.
Accessibility of tools	Requires narrowly specialized training of AltA facilitators and high-level analysts.	Democratization through no-code constructs (Condition, Loop, Function), accessibility for politicians and humanities professionals.	Expansion of the user base of strategic tools beyond technical experts.

Source: developed by the authors based on an analysis of the limitations of traditional methods [1, 2] and the capabilities of the proposed framework.

Conclusions

The proposed methodology of conflictological modeling, combining Alternative Analysis, semantic networking, and generative artificial intelligence, is not simply a sum of known tools, but constitutes a new methodological level, where critical thinking, formal structure, and computational power form a single system capable of constructing the future of conflicts. It is demonstrated that AltA, developed by NATO as a safeguard against groupthink, acquires new strength when combined with no-code prompt programming, where instead of the episodic application of techniques in workshops, a built-in mechanism for stress-testing plans emerges, operating automatically, scalably, and with a quantitatively verified result. Semantic networks, constructed based on entity and relation extraction, cease to be merely visualization – they become digital twins of conflicts, where each actor, event, motivation, and discursive frame obtains a formal representation suitable for analysis. Generative AI, in turn, functions not as an autonomous "expert", but as an intellectual service that implements AltA techniques in a reproducible, structured format: pre-mortem transforms into an iterative search for critical shift points, Team A/Team B – into parallel modeling of competing interests, and Surrogate Adversary – into a virtual embodiment of the opponent's logic without the risks of mirror-imaging. In this context, the human remains at the center of the process, but their role transforms: from an executor of routine cognitive operations to an architect of intellectual processes. This approach enables a transition from reactive analysis to proactive conflict management, where failure scenarios are modeled before they become viable, and points of tension are identified based on metrics of structural destabilization rather than by consequences.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Qwen to: translate certain text fragments from their native language, perform grammar and spelling checks, and paraphrase or

reword content. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Crowther, K.G. and Haimes, Y., 2008. Scenario Analysis, Cognitive Maps, and Concept Maps. Wiley Handbook of Science and Technology for Homeland Security

[2] da Silva, M.C.T., Times, V.C., de Macêdo, J.A. and Renso, C., 2015, October. SWOT: a conceptual data warehouse model for semantic trajectories. In Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP

[3] The NATO Alternative Analysis Handbook. Second Edition, December 2017. ISBN (PDF Version) 978-92-845-0208-0. URL: https://www.act.nato.int/wp-content/uploads/2023/05/alta-handbook.pdf

[4] Caballero, W.N. and Jenkins, P.R., 2025. On large language models in national security applications. Stat, 14(2), p.e70057

[5] Lande D., Strashnoy L. Advanced Semantic Networking based on the large language models : Monograph. Amazon Kindle Edition , 2025. 264 p. ISBN: 979-8297014190

[6] Hui, L., Wang, M., Zhang, L., Lu, L. and Cui, Y., 2022. Digital twin for networking: A data-driven performance modeling perspective. IEEE Network, 37(3), pp.202-209

[7] Arul, S.M., Senthil, G., Jayasudha, S., Alkhayyat, A., Azam, K. and Elangovan, R., 2023. Graph theory and algorithms for network analysis. In E3S Web of Conferences (Vol. 399, p. 08002). EDP Sciences.

[8] Hairol Anuar, S.H., Abas, Z.A., Yunos, N.M., Mukhtar, M.F., Setiadi, T. and Shibghatullah, A.S., 2024. Identifying Communities with Modularity Metric Using Louvain and Leiden Algorithms. Pertanika Journal of Science & Technology, 33(3)

[9] D.V. Lande, V.M. Furashev, Yu.G. Danyk, L.L. Strashnoy. Intelligent System of Parliamentary Control: From Semantic Analysis to Intelligent Coordination in Public Administration Using Artificial Intelligence : Monograph. – Kyiv: LLC "Inzhyniring", 2025. – 278 p.ISBN 978-617-8180-03-4 (Ukrainian language)