

LAMP

Understanding the Lockwood Analytic Method of Prediction

Nwankama Nwankama, PhD
Intelligence | Analytics

Understanding the Lockwood Analytical Method of Prediction (LAMP)

First published in 2010, revised in 2024

About the author

I am an intelligence and strategy professional with advanced training across engineering, technology, and organizational leadership. I earned a master's degree in Competitive Intelligence and Analytics from American Military University, where I studied under Col. Jonathan S. Lockwood, PhD (U.S. Army, Ret.), the creator of the Lockwood Analytical Method for Prediction (LAMP). Working under his supervision provided direct exposure to structured intelligence tradecraft grounded in real-world national security practice.

My academic foundation spans multiple disciplines. I previously completed master's degrees in architecture and engineering at McGill University in Montreal, following earlier graduate studies at the University of Nigeria. I later earned an additional master's degree in information technology from American Military University, with a focus on computer and intelligence systems, and completed a doctorate and specialized in Organizational Leadership and Change Management at Virginia University of Lynchburg.

Beyond formal education, I have pursued continuous professional development through executive and analytics programs with institutions such as the American College, Harvard Business School, and Stanford Graduate School of Business. My professional affiliations have included the American Academy of Management (Executive Member, where I reviewed scholarly articles), the Institute of Electrical and Electronics Engineers (IEEE)'s Computational Intelligence Society (New York), and the Harvard Business Review Advisory Council.

My work centers on producing actionable intelligence by integrating scientific, behavioral, and technological analytic methods to help organizations and leaders make sound decisions in both high-pressure and routine environments.

Table of Contents

Introduction	4
What is the LAMP?	5
Who Developed the LAMP?	6
What is the LAMP Process?	7
1) Defining the extrapolative issue	7
2) Specifying the actors bearing on the issue	8
3) Performing an in-depth study of the perceptions and intentions of each actor	8
4) Determining all possible courses of action for each actor	10
5) Determining the major scenarios within which the alternate futures are to be compared	10
6) Calculating the total number of permutations of possible "alternate futures" for each scenario	10
7) Performing a "pairwise comparison" of alternate futures	12
8) Ranking the alternate futures for each scenario from highest relative probability to the lowest based on the number of "votes" received	13
9) Analyzing each alternate future in terms of its consequences for the issue in question	13
10) Determining the "focal events" that must occur in the present time to bring about a given alternate future	14
11) Developing indicators for the focal events	15
12) Determining the potential of a given alternate future to "transpose" into another alternate future	15
Challenges Associated with LAMP	17
Challenge #1. Combinatorial Explosion of Alternate Futures	17
How computerization helps	17
Workbook overview (tabs)	22
1) Tab: Setup_Actors	22
2) Tab: Setup_Choices	22
3) Tab: Config	23
4) Tab: Futures_Generated	23
Core concept	23
Implementation note (so it's actually buildable)	24
5) Tab: Constraints (optional)	24
A) "Forbidden Pair" constraints (simple and powerful)	25

B) "Required Pair" constraints (optional)	25
6) Tab: Futures_Filtered	25
7) Tab: Pairwise_Queue	25
What we're building;	26
How to populate the queue (three sane options)	26
8) Tab: Export	27
A simple "minimum viable" version (if you want fast build)	27
Recommended "clean data rules" (avoid spreadsheet chaos)	27
Challenge # 2. Cognitive load and analyst fatigue	28
Challenge # 3. Subjectivity and analyst bias	28
Challenge # 4. Data quality and availability	28
Challenge #5. Time and resource intensity	29
Challenge #6. Communication of results	29
Current Use of LAMP and Its Practitioners	29
1. Intelligence and defense communities	29
2. Academic and professional training	30
3. Corporate and competitive intelligence	30
4. Policy and strategic planning support	30
Summary	30
Conclusion	31

Introduction

I was introduced to the Lockwood Analytical Method for Prediction (LAMP) not through a textbook, but directly by its creator, Col. Jonathan Lockwood, PhD—a retired U.S. Army Intelligence officer—while he served as Professor of Strategic Intelligence at American Military University, where I studied competitive intelligence and analytics.

Under his supervision, I applied LAMP to multiple analytic projects, gaining firsthand experience with the rigor, discipline, and intellectual honesty the method demands. That experience shaped how I approach intelligence problems: not as exercises in guesswork, but as structured examinations of human decision-making under uncertainty.

In today's strategic environment, leaders rarely fail because they lack data; they fail because they lack defensible insight into how adversaries, allies, and competitors may act. Traditional quantitative forecasting often breaks down when outcomes are driven by human choice rather than stable variables. LAMP was designed to address precisely this problem by treating the future as a competition among plausible alternatives shaped by actors exercising free will.

This booklet introduces LAMP to readers who may not all be intelligence professionals but who operate in national security, policy, or high-stakes strategic environments. It explains why the method works, where its challenges lie, and how modern tools can make it both scalable and practical. More importantly, it demonstrates how disciplined intelligence tradecraft—applied correctly—turns uncertainty into strategic advantage.

I hope you and your organization will benefit from the innovative processes that LAMP offers.

What is the LAMP?

The Lockwood Analytical Method for Prediction (LAMP) is a distinctive analytical framework designed to help intelligence analysts address the limitations of traditional quantitative forecasting techniques when assessing future events.

LAMP employs a blended predictive approach that draws upon several established analytical and planning methods, including the:

- Delphi Method
- Scenario Analysis
- The Analytic Hierarchy Process, and
- The Competing Hypotheses Method.

Rather than applying these tools in a purely numerical way, LAMP integrates key elements from each into a qualitative framework focused on comparing the relative likelihood of multiple alternative futures instead of calculating precise statistical probabilities.

Although LAMP does not assign numerical probabilities to individual outcomes, it minimizes ambiguity and produces clear, actionable insights. As a result, it offers strong intelligence support for policymakers engaged in decision-making.

LAMP works on the principle that the future is the sum total of all interactions of "free will," both on an individual, corporate and international scale.

In human history, the interactions of "free will" among actors have played out in predictable events.

While LAMP does not claim to predict the future with absolute certainty, it provides analysts with a more robust, structured, and dependable way to organize available information based on actor perceptions. By analyzing these perceptions through systematic comparison, alignment, and synthesis, LAMP enables analysts to judge which alternative future is most likely at a specific point in time. This, in turn, equips policymakers with reliable insights to develop effective strategic, operational, or tactical plans, taking into account multiple possible competitive responses.

Who Developed the LAMP?

The LAMP was developed in 1992-1993 by Dr. Jonathan Samuel Lockwood. He retired as a Colonel and Military Intelligence Officer in the US Army Reserves in May 2007. Dr. Lockwood developed the LAMP while he was a Fellow in the DCI “CIA Director” Analyst Program.

Dr. Lockwood holds a PhD in International Relations and an MS in Strategic Intelligence from the Joint Military Intelligence College. He also served as a Professor of Strategic Intelligence at American Military University, where he supervised some of my projects. He was also the Director of Training, Education, and Professional Development in the Office of Intelligence and Analysis in the United States Department of Homeland Security.

While Dr. Lockwood was the inventor of LAMP, I was the first person, on record, to start studying its potential applications in non-military and non-national security organizations, as we know them.

My attempt to apply a military conception to non-military use is not the first-time businesses, economic organizations, the military and security organizations would be learning from one another. The concept of strategic planning, which is discussed in corporate and organizational boardrooms today, was derived from the military. It was also the military, precisely, the U.S. Army, that worked with Bell Labs and major American universities in the 1940s, at the heat of World War II, to perfect the statistical sampling techniques that are widely used in business and organizational operations today. Again, the concepts of scenario planning and war gaming, which first emerged following World War II, as methods for military planning are widely applied in businesses and organizations today.

In turn, when the United States defense planning body found itself at a crossroads, they found it imperative to review not only discrete issues, but even basic concepts of strategic planning. They turned to businesses and civilian organizations.

Bracken (1990) reviewed concepts that the military and US security agencies would examine, which were used in the civilian world to build insights about how to view current and prospective problems, opportunities, and choices within the defense enclave. The author saw the military draw upon business and civilian literature for both ideas and metaphors. He surmised that U.S. national security planning could profit greatly from an approach that distinguished among “core, environmental, and hedging strategies,” and that considered security analogs to such business and organizational leadership concepts as defining the

business or organization, dealing with new competitors, controlling the intensity of competition, entry and exit barriers, and the need to redeploy assets and restructure the organization. He then related these concepts to specific problems of national security interest.

What is the LAMP Process?

The LAMP is a 12-step meticulous (and somewhat technical) process, which defines scopes and analyzes an issue of a very serious strategic, operational or tactical importance.

The LAMP process goes as follows:

STEP

01

Defining the extrapolative issue

The analyst, in concert with decision makers in the applicable organization, must carefully decide what exact predictive issue he or she wants to confront. This is an extremely important step in the LAMP method.

If the extrapolative issue is too broad, the analyst will have too many actors and courses of action to consider. This will cause the number of alternate futures to explode exponentially. If on the other hand, the issue is vaguely defined, then the analyst will end up conducting a study that is too general. This will make the subsequent comparisons of alternate futures less reliable and will result in a routine collection of conjectures – that are hardly actionable if the strategic, operational or tactical matter is serious.

STEP



Specifying the actors bearing on the issue

The analyst should then determine the number of "actors" who can directly affect the predictive issue. If the analyst is careful enough to limit the scope of his or her initial question, the number of "actors" involved should be no more than 5 or 6.

Depending on the predictive issue in question, the "actors" can be individuals, group of persons or countries. If the number of "actors" is more than 6, the number of possible "alternate futures" may become unmanageable. Unless the analyst has programmed the specific organization's LAMP template into computer software, it will not be easy to handle the large number of permutations that will arise.

STEP



Performing an in-depth study of the perceptions and intentions of each actor

This step involves the greatest amount of historical research for the analyst, and is the most time-consuming. Not only should the analyst examine current history from the actor's viewpoint, but he or she should also look for historical events, cultural factors, and nuances of language that might have an impact on an actor's or subject's outlook. Here, it would be germane to apply the intelligence technique of Words of Estimative Probability – whereby the analyst would, on separate worksheets, assign numerical odds to key phrases that have been used by the actors in the predictive issue in question.

Failure to conduct an adequate perceptual study increases the danger that the analyst will fall into the "mirror-imaging" trap of substituting his or her own logic for the actor's. If this occurs, it will skew the analyst's calculations when comparing the likelihood of the various "alternate futures," ultimately yielding less reliable intelligence.

The analyst should be well-informed that credibility is paramount in intelligence analysis. Renowned US intelligence master – Sherman Kent's original process

in the formulation of intelligence estimates hinges on credibility of the intelligence, and in recognizing the serious nature of the estimate.

Credibility requires that those charged with preparing intelligence estimates remain sound. Sherman Kent stressed that an estimate should not only be relevant within the area of the competence of the intelligence professional, but that it should be credible. In fact, the process should be such that intelligence workers should be so credible that if the policymaking master is to disregard the knowledge and wisdom of the intelligence professional, he will never do so because the intelligence work was inaccurate, incomplete, or patently biased. "Let him disregard us only when he must pay greater heed to someone else. And let him be uncomfortable—thoroughly uncomfortable—about his decision to heed this other."¹

Sherman Kent (1903–1986), the father of CIA intelligence analysis, argued that intelligence estimates must be so rigorous, objective, and credible that policymakers who reject them will do so, only with extreme discomfort—not due to bias or poor analysis.



So, the key issues here are for the analyst to be exhaustive in his or her perceptual study and to not interject his or her opinions or preferences in sorting the perceptions and intentions of any actor.

¹ Sherman Kent and the Board of National Estimates, "Estimates and Influence" in 'Collected Essays', - a classic exposition of estimative intelligence, which treats both its epistemology and its importance to the policymaker. This was classified Confidential and published in the Summer 1968 number of Studies of Intelligence.

STEP

04

Determining all possible courses of action for each actor

After completing the perceptual study of each actor, the possible courses of action should become apparent to the analyst.

The analyst should not exclude a course of action merely because it seems unlikely that an actor will choose it. However, choices that are clearly impossible or absolutely beyond the power of the actor in question should be excluded.

STEP

05

Determining the major scenarios within which the alternate futures are to be compared

One purpose of a scenario is to provide the major assumption which influences the actions of all actors concerned. Often it is based on the actions of a major power outside the scope of an analyst's initial study.

The other purpose, equally as important, is to give the analyst a means through which he or she can limit the potential number of actors for the problem.

STEP

06

Calculating the total number of permutations of possible "alternate futures" for each scenario

The analyst should remember that in several fields of mathematics, the term "permutation" is used with different but closely related meanings. They all relate to the notion of mapping the elements of a set to other elements of the same set, i.e., exchanging (or "permuting") elements of a set.

A formula for the number of possible permutations of k objects from a set of n is usually written ${}_nP_k$.

Thus;

$$P(n, k) = \frac{n!}{(n - k)!}$$

For this, and the next step, the analyst should have, at least, some basic knowledge of calculus. For our purpose here, permutations and combinations each have very specific meanings, and the analyst should remember that this distinction often causes problems if not handled with diligence.

In brief, the permutation of a number of objects (futures, in own case) is the number of different ways they can be ordered; i.e. which is first, second, third, etc. If the analyst wishes to choose some futures from a larger number of futures, the way he or she positions the chosen futures is also important. With combinations, on the other hand, the analyst does not consider the order in which futures were chosen or placed, just which futures were chosen.

Permutations and combinations (very simplistically) can be summarized as:

Permutations - Position important (although choice may also be important) **Combinations - Chosen important.**

This may help the analyst to remember which is which.

Here is where the necessity of limiting the number of actors and choices becomes most apparent. The simple formula for computing the number of alternate futures is:

$$X^Y = Z$$

Where X equals the number of courses of action open to each actor, and Y equals the number of actors involved (assuming each actor has the same number of courses of action open to it). Z equals the total number of alternate futures to be compared. For example, if the analyst is looking at five actors with two courses of action open to each, then he or she is looking at only 32 (note, not 10 or 25) alternate futures. If however, the analyst includes another actor with three courses of action, then the number of alternate futures becomes 96. However, if there were three courses of action open to all five actors, then the number of possible alternate futures would explode to 243.

Obviously, the use of scenarios helps the analyst keep the problem within manageable bounds. It should be noted in this step, that for scenarios to be effective, they should be modeled as closely as possible on the existing world, or if departures are made, the changes should be explained as completely as possible. In some cases, relevancy rather than credibility should predominate. It is important to know that scenarios are less than half-way through a LAMP analysis.

The aim of scenarios is not to foresee the future, but to show how different interpretations of the driving forces of change can lead to different possible futures.

STEP

07

Performing a “pairwise comparison” of alternate futures

In this step, the analyst will perform a "pairwise comparison" of all alternate futures within the scenario to determine their relative probability. This comparison will analyze the alternate futures two at a time, always assuming that the two futures being compared at the moment are the only ones that exist. Based on all of the information the analyst is aware of at that moment, whichever future is deemed "more likely to occur" is given one "vote."

Each future is compared to all futures within the scenario. This continues until the analyst has compared and "voted" on the last pair of futures. This process repeats for the other scenarios.

The total number of "votes" is a function of the number of alternate futures to be analyzed, which in turn is a function of the number of actors and courses of action the analyst has determined for the issue.

The formula for the number of "pairwise comparison" is:

$$X = \frac{n(n-1)}{2}$$

Where n equals the total number of alternate futures to be analyzed, and X equals the total number of "pairwise comparisons."

However, as the number of actors and courses of action involved is increased, it becomes apparent how quickly the size of the problem can become unmanageable without either computer support or a prior decision on the part of the analyst to limit the scope of the analysis. As the analyst proceeds through the "pairwise comparison", he or she will discover that some "pairs" are easier to "vote" on than others.

STEP

08

Ranking the alternate futures for each scenario from highest relative probability to the lowest based on the number of "votes" received

The "pairwise comparison" yields a series of futures receiving different numbers of "votes" based upon their relative probability to each other. The analyst then rank-orders the futures from "most likely" to "least likely" based upon the number of "votes" received.

STEP

09

Analyzing each alternate future in terms of its consequences for the issue in question

In this step, the analyst must assume that each future will occur. It will require some vision and imagination on the part of the analyst. He or she is, after all, writing "future history" of things that might be, given that the actors take the courses of action of a particular alternate future.

Depending on the predictive issue and the degree of research the analyst is willing to undertake, describing the consequences of a given alternate future often takes longer than a few paragraphs.

In this step, like in Step 3, the analysts must be very conscious of the ethos of his or her profession as an intelligence analyst. He or she has embarked on reducing the ambiguity of a highly ambiguous situation. His or her task is made more challenging because of the likelihood of counterintelligence efforts by people on the other side of the issue, or because the ambiguity is very

deliberately created by highly intelligent actors who have mindsets that are very different from the analyst's outlook.

Many analysts prefer the middle-of-the-road explanation, rejecting high or low probability explanations. However, whatever the analyst's philosophies, experience or persuasion, he or she must avoid the special cognitive traps for intelligence analysis – especially the traps of his or her own personality, or of the analyst's organizational culture – projecting what she or he (or his or her organization) wants the opponent to think, and using available information to justify that conclusion.

Again, intelligence needs to be neither optimistic nor pessimistic. It should be realistic.



Determining the "focal events" that must occur in the present time to bring about a given alternate future

A "focal event" is an occurrence of sufficient magnitude that it changes the relative probability of an alternate future. That is, once the path is taken, there would be a different array of possible futures and branching off points. The most likely future will have the fewest focal events leading into it. That means that it will be the alternate future offering the line of least resistance.

The more "bizarre" futures, on the other hand, would have more focal events leading to them, since more of such events are required in order to change the present into those futures. These would be the alternate futures receiving the fewest "votes" during the "pairwise comparison" in Step 7.

STEP

11

Developing indicators for the focal events

As the next-to-last step in the LAMP process, this step would link the LAMP technique with the more familiar Indications and Warning process. For each focal event associated with an alternate future, it should be possible to develop a list of indicators that such an event either would have already occurred or was about to occur.

Once those indicators are entered into an automated database along with the focal events and alternate futures, the analyst has almost completed the basic 12-step LAMP process. Subsequent activity would consist of periodic "revoting" of the alternate futures as new information is acquired, refinement of the indicators associated with particular focal events, as well as the identification of additional focal events for the more exotic alternate futures.

STEP

12

Determining the potential of a given alternate future to "transpose" into another alternate future

In informal language, a transposition is a function that swaps two elements of a set. The concept of "transposition" is well-known in analytics. Yet, an analogy will later be drawn from the common game of Chess, where a player may arrive at a position using a different sequence of moves; or sometimes where he or she deliberately moves in order to avoid variations they dislike, lure opponents into unfamiliar or uncomfortable territory or just to worry opponents.

For instance, the first position can be obtained from the Queen's Gambit:

1. d4 d5
2. c4 e6
3. Nc3 Nf6

But this position can also be reached from the English opening:

1. c4 Nf6

2. Nc3 e6
3. d4 d5

So, the English opening has transposed into the Queen's Gambit.

More formally, given a finite set:

$$X = \{a_1, a_2, \dots, a_n\}$$

That means that a transposition is a permutation (bijective function of X onto itself) f , such that there exist indices i, j such that $f(a_i) = a_j$, $f(a_j) = a_i$ and $f(a_k) = a_k$ for all other indices k . This is often denoted (in the cycle notation) as (a, b) .

We can surely see transposition when we consider the following scenario;

If $X = \{a, b, c, d, e\}$ the function σ given by:

$$\begin{aligned}\sigma(a) &= a \\ \sigma(b) &= e \\ \sigma(c) &= c \\ \sigma(d) &= d \\ \sigma(e) &= b\end{aligned}$$

The above scenario is an example of a transposition.

The intelligence analyst should remember that any permutation can be expressed as the composition (product) of transpositions. Again, the analyst should remember that one of the main results on symmetric groups states that either all of the decompositions of a given permutation into transpositions have an even number of transpositions, or they all have an odd number of transpositions.

Back to Chess-playing Opening Theory, transposition occurs when one chess opening's line of play leads into one resembling a different chess opening. The same occurs in the LAMP's array of alternate futures. Since every act of "free will" changes the future, it has virtually the same result as transposition does in chess.

Once the analyst has described the consequences of an alternate future, its potential for transposition into another alternate future should be noted, since this may affect the relative probability of those futures.

Challenges associated with LAMP

While LAMP is a powerful and systematic methodology, over the years, I have seen that it presents several practical and methodological challenges. I have also worked to address some of those challenges.

The challenges are as follows:

1. Combinatorial Explosion of Alternate Futures

One of the most significant challenges with LAMP is the rapid growth in the number of possible alternate futures as the number of actors and choices increases. Because the total number of futures is calculated as c^a (choices raised to the number of actors), even modest increases can produce hundreds or thousands of permutations.

This can quickly overwhelm analysts, making manual comparison impractical.

How computerization helps

I have used software tools, spreadsheets, and specialized decision-support systems to automate permutation generation, ranking, and comparison, significantly reducing analyst workload and error.

What you want the tool to do (requirements that matter)

A. *Generate alternate futures automatically*

Minimum inputs:

- **Actors** (A1...An)
- **Choices per actor** (C1...Cm, can be actor-specific)
- Optional: **constraints** (“A1 can’t choose C3 if A2 chooses C1”, mutual exclusivity, sequencing, etc.)

Outputs:

- A generated list of futures (rows), each future = one choice per actor.
- Counts: total futures, and counts after constraints.

This is a pure enumeration / constraint-satisfaction problem (easy for software; painful for humans).

B. Make ranking scalable

This is important so you don't do impossible numbers of comparisons.

LAMP often relies on comparing futures to decide which is “more likely.” If you try to compare *all futures to all others*, it exponentially explodes.

So, a good tool is one that supports at least one of these strategies:

1. **Pairwise comparisons with built-in logic checks**
Borrow from AHP-style pairwise comparison workflows, including **consistency checks** (helps catch “I said A>B and B>C but also C>A” kinds of judgment loops). This is a mature software niche.
2. **Tournament / bracket ranking**
Instead of comparing every future to every other, the system presents a smart subset of comparisons, “advancing” winners. (Not perfect, but fast and often good enough operationally.)
3. **Criteria-based scoring + sensitivity**
Translate “likelihood” into a weighted model (e.g., actor capability, intent, constraints, incentives), compute a score, then do sensitivity analysis.

C. Reduce workload and error with guardrails

Look for:

- **Audit trail** (who entered what judgment, when)
- **Assumption log** (what changed between iterations)
- **Consistency diagnostics** (flag illogical judgments)
- **Versioning** (so you can re-run when new intel arrives)

Three practical tool paths (from “simple” to “serious”)

1) Spreadsheet-first (fastest to deploy)

Design pattern in Excel/Google Sheets

- Sheet 1: Actors + choices (including actor-specific choice lists)
- Sheet 2: Enumerated futures (auto-generated via formulas / PowerQuery / Apps Script / VBA)
- Sheet 3: Comparison inputs (pairwise prompts or criteria weights)
- Sheet 4: Ranking + charts + sensitivity

When it works best: small-to-medium problems, rapid iteration, teams already live in Excel.

How to add “decision-support” power inside Excel

- Use established Excel add-ins built for decision analysis and uncertainty:
 - **Palisade DecisionTools Suite** (includes **PrecisionTree** for decision trees + **@RISK** for Monte Carlo-style uncertainty work inside Excel).

This won’t “do LAMP” out of the box, but it can handle structured decision modeling, uncertainty, and scenario comparison in the same spreadsheet environment.

2) Use AHP/pairwise-comparison software to handle ranking cleanly

This is often the sweet spot because LAMP ranking is conceptually close to pairwise preference/ranking workflows.

Options to evaluate:

- **Expert Choice (Companion)** — purpose-built AHP-style structuring and pairwise judgments.
- **TransparentChoice** — emphasizes streamlined pairwise comparisons, **built-in consistency checks**, and visualization.
- **SuperDecisions** — supports hierarchical pairwise comparison models and sensitivity analysis (AHP/ANP style).
- **1000minds (PAPRIKA)** — designed around *pairwise rankings of alternatives* and reducing the burden of comparisons.

How you'd use these with LAMP

- You still generate the futures (Excel or a simple script).

- Then import the futures as “alternatives” into one of the above tools.
- Define “likelihood drivers” as criteria (if you want criteria-based ranking) or do direct pairwise comparisons of futures (for smaller sets).
- Use the tool’s consistency/sensitivity features to keep judgments coherent and defensible.

3) Build or adopt a specialized decision-support app

This is best for large problems.

If you routinely face large actor/choice spaces or lots of constraints, you eventually want:

- A **constraint engine** (rule-based pruning of impossible futures)
- A **database** of actors/choices/assumptions
- A **workflow UI** for comparisons + collaboration
- Automated **reports** (top futures + “why” narratives)

As, so far, you can’t find “LAMP software” marketed as such, this is exactly the kind of functionality decision-support platforms are built to provide—your differentiator is the LAMP-specific data model and reporting.

A concrete “evaluation checklist” for existing tools

When I test candidates (spreadsheet add-in, AHP tool, or custom app), these are the questions I ask:

1. **Can it import/export futures easily?** (CSV in/out is non-negotiable)
2. **Does it support constraints or at least filtering?**
3. **Does it reduce comparison burden?** (smart sampling, partial pairwise, etc.)
4. **Does it detect inconsistency/provide diagnostics?**
5. **Can it show sensitivity (“if assumption X changes, what happens”)?**
6. **Collaboration:** multiple analysts, review/approval, audit trail
7. **Explainability:** can it generate a “top 5 futures + drivers” brief?

A sensible hybrid that works in practice

A lot of teams land on this because it's efficient and defensible:

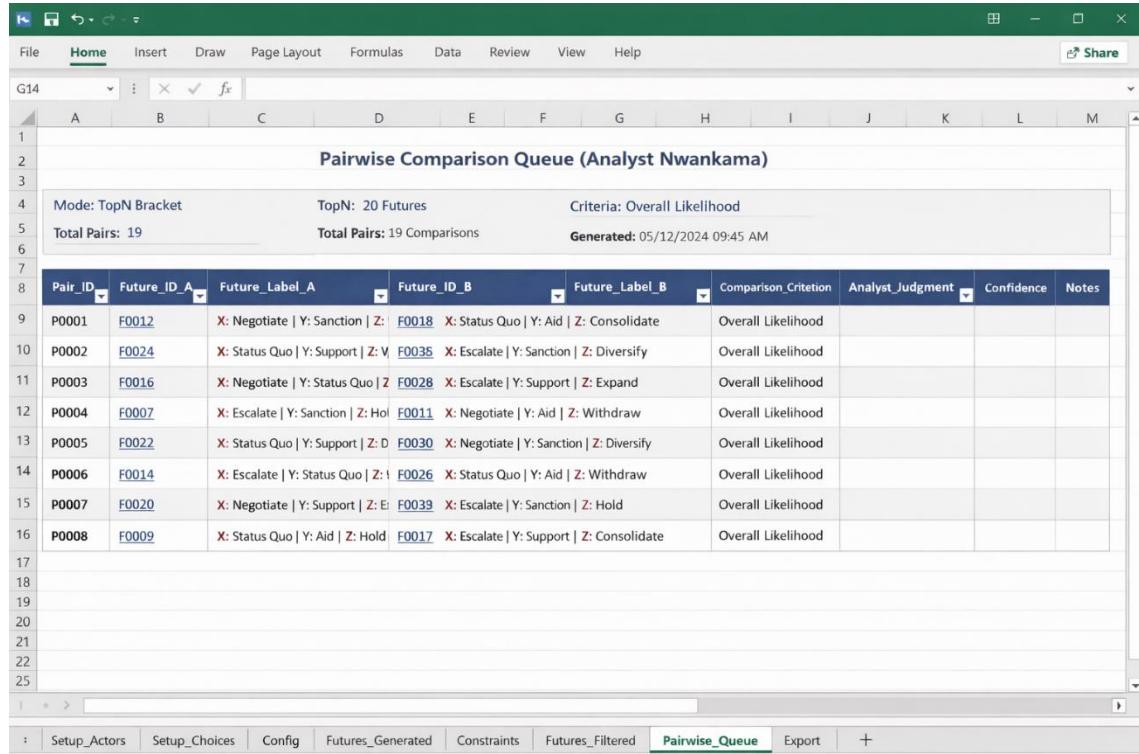
- **Excel (or a small script) for generation + constraints**
- **AHP/pairwise tool for ranking + consistency checks**
- **Excel/PowerPoint outputs for communication**

That setup directly targets the LAMP pain point: explosion of futures + human inconsistency.

An Excel schema

Here, I have sketched a clean Excel schema (tabs + column layouts) that generates futures and prepares a “pairwise comparison queue” that can be fed into an AHP/pairwise tool.

Here's a screenshot of one I built:



Pairwise Comparison Queue (Analyst Nwankama)									
Mode: TopN Bracket			TopN: 20 Futures			Criteria: Overall Likelihood			
Total Pairs: 19			Total Pairs: 19 Comparisons			Generated: 05/12/2024 09:45 AM			
Pair_ID	Future_ID_A	Future_Label_A	Future_ID_B	Future_Label_B	Comparison_Criterion	Analyst_Judgment	Confidence	Notes	
P0001	F0012	X: Negotiate Y: Sanction Z: F0018	X: Status Quo Y: Aid Z: Consolidate		Overall Likelihood				
P0002	F0024	X: Status Quo Y: Support Z: V	F0035 X: Escalate Y: Sanction Z: Diversify		Overall Likelihood				
P0003	F0016	X: Negotiate Y: Status Quo Z: F0028	X: Escalate Y: Support Z: Expand		Overall Likelihood				
P0004	F0007	X: Escalate Y: Sanction Z: Hol	F0011 X: Negotiate Y: Aid Z: Withdraw		Overall Likelihood				
P0005	F0022	X: Status Quo Y: Support Z: D	F0030 X: Negotiate Y: Sanction Z: Diversify		Overall Likelihood				
P0006	F0014	X: Escalate Y: Status Quo Z: I	F0026 X: Status Quo Y: Aid Z: Withdraw		Overall Likelihood				
P0007	F0020	X: Negotiate Y: Support Z: E	F0039 X: Escalate Y: Sanction Z: Hold		Overall Likelihood				
P0008	F0009	X: Status Quo Y: Aid Z: Hold	F0017 X: Escalate Y: Support Z: Consolidate		Overall Likelihood				

A screenshot of an Excel schema that generates futures and prepares a “pairwise comparison queue” that I can feed into an AHP/Pairwise tool. Click here to view a larger image of the screenshot online

This clean, “works in real life” Excel schema can be built that:

- (1) Defines actors/choices
- (2) Generates alternate futures, and
- (3) Outputs a pairwise-comparison queue you can paste/import into an AHP / pairwise tool.

I assumed that **each actor selects exactly one choice** in a future. I also included a simple way to add constraints (optional) without making the workbook fragile.

Workbook overview (tabs)

1. Setup_Actors
2. Setup_Choices
3. Config
4. Futures_Generated
5. Constraints (*optional but recommended*)
6. Futures_Filtered
7. Pairwise_Queue
8. Export

1) Tab: Setup_Actors

Purpose: define the actors and their metadata.

Columns

- A: Actor_ID (A1, A2, A3...)
- B: Actor_Name (e.g., “Country X”, “CEO”, “Militia Y”)
- C: Actor_Type (Individual/Org/State/etc.)
- D: Notes

Example:

Actor_ID	Actor_Name	Actor_Type	Notes
A1	Country X	State	election in 6 months
A2	Neighbor Y	State	security concerns
A3	Firm Z	Org	supply-chain exposure

2) Tab: Setup_Choices

Purpose: list the available choices, actor by actor (actor-specific choice sets).

Columns

- A: Actor_ID (must match Setup_Actors)
- B: Choice_ID (C1, C2... OR actor-specific like A1_C1)
- C: Choice_Label (short: “Escalate”, “Sanction”, “Negotiate”)
- D: Choice_Description (optional)
- E: Active (TRUE/FALSE) (*lets you toggle choices without deleting*)

Example

Actor_ID	Choice_ID	Choice_Label	Choice_Description	Active
A1	A1_C1	Escalate	increase troop presence	TRUE
A1	A1_C2	Negotiate	offer talks via UN	TRUE
A1	A1_C3	Status quo	no change	TRUE
A2	A2_C1	Sanction	target finance	TRUE
A2	A2_C2	Support	provide aid	TRUE

3) Tab: Config

Purpose: central settings and helper ranges.

Cells

- B1: Max_Futures (hard cap; e.g., 5000 to prevent blowups)
- B2: Pairwise_Mode (“TopN”, “RandomK”, “AllPairsUnderN”)
- B3: TopN (e.g., 50)
- B4: RandomK (e.g., 200)
- B5: Seed (optional for repeatable random sampling)
- B6: Include_Constraints (TRUE/FALSE)

Helper tables

- A small dynamic list of **Active Actors** and **Active Choices per actor** (used by formulas).

4) Tab: Futures_Generated

Purpose: enumerate all combinations (alternate futures) up to Max_Futures.

Core concept

Each row = one future. Each actor gets one choice.

Columns

- A: Future_ID (F0001, F0002...)
- B..(B+n-1): Choice_for_A1, Choice_for_A2, ... (one column per actor)
- Next: Future_Label (human-readable concatenation)
- Next: Base_Score (*optional placeholder; could stay blank until later*)

Example columns for 3 actors

Future_ID	A1_Choice	A2_Choice	A3_Choice	Future_Label
F0001	A1_C1	A2_C1	A3_C1	A1:Escalate A2:Sanction A3:...
F0002	A1_C1	A2_C1	A3_C2	...

Implementation note (so it's actually buildable)

To me, the easiest *reliable* way in Excel is:

- **Power Query** to cross-join choices across actors, or
- **Office Scripts/VBA** to generate rows.

Caution:

Some analysts may want *formula-only*. It's possible but gets ugly fast when actors have uneven choice counts. In practice, most teams use Power Query because it's built-in and auditable.

Power Query approach (recommended)

- Load Setup_Choices as a table.
- Filter Active=TRUE.
- Split by Actor_ID into separate queries (A1 choices, A2 choices...).
- Cross-join them into a full futures table.
- Load the result back into Futures_Generated.

This gives you clean regeneration when you toggle choices.

5) Tab: Constraints (optional)

Purpose: encode simple “invalid futures” rules (pruning).

Two constraint styles that work well in Excel:

A) “Forbidden Pair” constraints (simple and powerful)

Invalid if (Actor i chooses X) AND (Actor j chooses Y).

Columns

- A: Constraint_ID
- B: Actor_ID_1
- C: Choice_ID_1
- D: Actor_ID_2
- E: Choice_ID_2
- F: Reason
- G: Active (TRUE/FALSE)

B) “Required Pair” constraints (optional)

If actor chooses X, another actor must choose from {Y1,Y2...}. (Harder; skip unless needed.)

6) Tab: Futures_Filtered

Purpose: bring forward generated futures + flag validity + keep only valid ones.

Columns

- All columns from Futures_Generated
- Validity_Flag (VALID/INVALID)
- Invalid_Reasons (optional)

How it works

For each forbidden pair constraint, test whether the future row matches it. If any match, mark INVALID.

(If you implement with Power Query, you can also do constraint filtering in PQ; otherwise a formula-based flag works.)

7) Tab: Pairwise_Queue

Purpose: create a list of future pairs for analysts to compare, or for export into AHP/pairwise software.

What we're building;

A table with rows like: "Compare Future_A vs Future_B".

Columns

- A: Pair_ID (P000001...)
- B: Future_ID_A
- C: Future_Label_A
- D: Future_ID_B
- E: Future_Label_B
- F: Comparison_Criterion (*optional: "Overall likelihood" or criterion name*)
- G: Analyst_Judgment (*blank: A/B/Tie or 1-9 scale*)
- H: Confidence (*optional: Low/Med/High or 1-5*)
- I: Notes

How to populate the queue (three sane options)

Option 1 — “TopN bracket” (best workload control)

- Take the first TopN valid futures (or later, TopN after a coarse pre-score).
- Generate a structured set of comparisons:
 - Round 1: pair 1 vs 2, 3 vs 4, etc.
 - Round 2: winners compared, etc.This is very manageable for humans.

Option 2 — “RandomK comparisons” (good for big sets)

- Randomly sample K pairs from the valid futures set.
- Useful when futures are too many and you want probabilistic coverage.

Option 3 — “All pairs if small” (only if $N \leq \sim 30$)

- If N is small, you can generate all unique pairs: $N*(N-1)/2$.

In **Config**, you select mode and the sheet generates the queue accordingly.

8) Tab: Export

Purpose: create the clean files you'll copy/paste or save as CSV into the AHP tool.

Exports

- `Futures_For_Tool.csv` (`Future_ID`, `Future_Label`)
- `Pairs_For_Tool.csv` (`Pair_ID`, `Future_ID_A`, `Future_ID_B`, `Criterion`)

Some tools want the pairwise judgments to be collected in their interface; others allow importing a “comparison queue” template. This export tab keeps formatting stable.

A simple “minimum viable” version (if you want fast build)

If you want to build this in an afternoon:

- Use **Power Query** to generate `Futures_Generated` from `Setup_Choices`.
- Skip constraints at first (or do a simple VALID flag later).
- Use `Pairwise_Queue` with **TopN** + “adjacent pairing” (1v2, 3v4...).

That alone removes the biggest pain: manual enumeration and ad hoc pairing.

Recommended “clean data rules” (avoid spreadsheet chaos)

- Make every setup range a proper **Excel Table**:
 - `tblActors`, `tblChoices`, `tblConstraints`, `tblFutures`, `tblPairs`
- Never hardcode Actor columns except in `Futures_Generated` (which is output).
- Keep IDs machine-friendly (no spaces), keep labels human-friendly.
- Add a **Last_Updated** timestamp in Config when you refresh PQ.

Challenge # 2. Cognitive Load and Analyst Fatigue

LAMP requires analysts to repeatedly compare futures pairwise to determine relative likelihoods. This is cognitively demanding and time-intensive, increasing the risk of inconsistency, fatigue, or bias—especially in large models.

Mitigation:

Structured software interfaces, automated consistency checks, and collaborative team-based analysis can help distribute the workload and improve reliability.

Challenge # 3. Subjectivity and Analyst Bias

LAMP is intentionally qualitative and relies heavily on analyst judgment, particularly in assessing actor perceptions and preferences. While this is a strength, it also introduces the risk of personal, organizational, or cultural bias influencing outcomes.

Mitigation:

Red teaming, peer review, structured elicitation techniques, and explicit documentation of assumptions can help expose and reduce bias.

Challenge # 4. Data Quality and Availability

The accuracy of LAMP outputs depends on the quality of information about actors, their motivations, and their perceived options. In many intelligence contexts, such information is incomplete, deceptive, or rapidly changing.

Mitigation:

LAMP works best when continuously updated; computerization enables rapid revisions as new intelligence becomes available.

Challenge #5. Time and Resource Intensity

Conducting all 12 steps of LAMP thoroughly can be time-consuming, making it less suitable for fast-breaking crises or environments requiring immediate decisions.

Mitigation:

Analysts may use a “truncated” or focused version of LAMP for tactical situations, supported by pre-built actor models and databases.

Challenge #6. Communication of Results

Because LAMP does not produce numerical probabilities, some decision-makers accustomed to quantitative metrics may initially struggle to interpret or trust its findings.

Mitigation:

Visualization tools (ranked futures, matrices, and decision trees) and clear narrative explanations help translate results into actionable insights.

Current Use of LAMP and Its Practitioners

1. Intelligence and Defense Communities

LAMP is most commonly used within intelligence, military, and national security organizations, particularly in the United States.

It has been applied to:

- Strategic forecasting
- Foreign policy analysis
- Military planning and wargaming
- Counterterrorism and geopolitical risk assessment

It is especially valued in environments where human decision-making and strategic interaction drive outcomes.

2. Academic and Professional Training

LAMP is taught in intelligence studies programs, military education institutions, and analytic tradecraft courses. It is often used as a case-study-based teaching tool to train analysts in structured thinking, alternative futures analysis, and bias mitigation.

3. Corporate and Competitive Intelligence

Some private-sector organizations and consultants use LAMP or LAMP-inspired approaches for:

- Competitive strategy
- Market entry analysis
- Corporate risk assessment
- Long-term strategic planning

Its focus on actor perceptions and competitive response makes it particularly useful in adversarial or oligopolistic environments.

4. Policy and Strategic Planning Support

Think tanks and advisory groups employ LAMP to provide policymakers with structured foresight on complex issues such as international negotiations, sanctions regimes, or alliance dynamics. Its qualitative clarity helps leaders explore “what if” scenarios without relying solely on uncertain quantitative forecasts.

Summary

LAMP’s greatest strength—its structured, qualitative comparison of alternate futures driven by human free will—is also the source of its main challenges: scale, subjectivity, and resource demands. Advances in computerization,

collaboration tools, and analytic software have significantly improved its practicality.

Today, LAMP remains a respected and widely used methodology among intelligence professionals, strategists, and planners who must operate in complex, competitive, and uncertain environments.

Conclusion

LAMP's enduring strength lies not in its promise of certainty, but in its insistence on analytical honesty. By compelling analysts to explicitly model actors, choices, and perceptions, it transforms intuition into structured judgment and exposes assumptions that would otherwise remain hidden.

The methodological challenges—particularly the rapid growth of alternate futures—are not weaknesses of LAMP, but reflections of the real complexity of human decision-making. When paired with modern tools that automate generation, comparison, and ranking, those challenges become manageable and analytically productive.

My experience learning LAMP under the direct guidance of its creator, Dr. Jonathan Lockwood, reinforced a core principle of intelligence professionalism: credible analysis does not tell policymakers what they want to hear—it gives them a defensible understanding of what *may* happen and why. In this sense, LAMP stands squarely in the tradition championed by Sherman Kent: intelligence that can be rejected, but never dismissed as careless, incomplete, or biased.

For leaders operating in contested environments, and for analysts tasked with informing them, LAMP offers something increasingly rare—a rigorous way to think about the future without pretending it is predictable. Mastery of this method signals more than technical competence; it demonstrates disciplined strategic thinking under uncertainty. In a world where decisions are made with imperfect information and real consequences, that capability is not simply valuable—it is indispensable.

References

Austin, J., & Delaney, P. F., "Protocol Analysis as a Tool for Behavior Analysis" in "Analysis of Verbal Behavior" Volume 15, 1998, pp 41-56.

Bracken, P., "Strategic Planning for National Security: Lessons from Business Experience", Rand Corporation, Santa Monica, CA, Feb 1990

Crutcher, R. J., "Telling What We Know: The Use of Verbal Report Methodologies in Psychological Research", Psychological Science, Volume 5, 1994, pp 241-244.

Ericsson, A.K. & Simon, H.A., "Protocol Analysis: Verbal Reports as Data", MIT Press, Cambridge, MA, 1993.

Ericsson, K. A., & Crutcher, R. J., "Introspection and Verbal Reports on Cognitive Processes - Two Approaches to the Study of Thought Processes: A response to Howe" in New Ideas in Psychology, Volume 9, 1991, pp. 57-71.

Pressley, M., & Afflerbach, P., "Verbal Protocols of Reading: The Nature of Constructively Responsive Reading", Erlbaum, 1995

Renkl, A., "Learning From Worked-out Examples: A Study on Individual Differences" in Cognitive Science, Volume 21, 1997, pp. 1-29.

Sudman, S., Bradburn, N. M., & Schwarz, N. (Eds.), "Thinking About Answers: The Application of Cognitive Processes to Survey Methodology", Jossey-Bass, 1996.