# 系統性湧現：構建基於效用AI、黑板架構與ABM的自模擬政治經濟遊戲世界

## 1. 引言：從腳本敘事到系統性湧現的典範轉移

在數位互動娛樂與模擬技術的演進歷程中，創造一個具有「生命力」的虛擬世界始終是開發者與研究人員追求的終極目標。傳統的遊戲世界設計往往受限於「腳本化」的思維模式，即經濟波動、政治變革或社會動盪皆由設計師預先編寫的觸發器（Triggers）所控制。這種由上而下的設計方法（Top-Down Design）雖然能夠確保敘事的連貫性，卻犧牲了世界的動態響應能力與玩家的沉浸感。當玩家試圖干預一個貿易路線時，除非設計師預先撰寫了相關的後果腳本，否則世界往往無動於衷，這種現象被稱為「靜態背景」問題。為了突破這一瓶頸，當代遊戲AI與模擬架構正經歷一場深刻的典範轉移，即轉向**系統性遊戲設計（Systemic Game Design）**。

系統性設計的核心理念在於，宏觀的複雜現象不應被直接編寫，而應是微觀代理（Agents）在特定規則下互動的數學總和，即所謂的**湧現行為（Emergent Behavior）** 1。在這種架構下，戰爭不是由腳本觸發的事件，而是資源匱乏、外交失敗與機會主義計算的必然結果；飢荒不是隨機數生成的狀態，而是物流中斷、氣候變遷與市場囤積行為的連鎖反應。要實現這樣一個自運轉、自洽且具備高度可信度的虛擬社會，必須整合多種先進的計算方法：用於個體決策的**效用人工智慧（Utility AI）**、用於全局資訊協調的**黑板系統（Blackboard Systems）**、用於模擬經濟流動的**基於代理的模型（Agent-Based Modeling, ABM）**、用於處理外交博弈的**賽局理論（Game Theory）**，以及用於模擬訊息傳遞失真的**資訊擴散演算法（Information Diffusion Algorithms）**。

本研究報告將深入探討上述技術的整合架構，旨在為設計一個能自行模擬經濟供需與政治演化的驅動型遊戲世界提供詳盡的理論基礎與實踐藍圖。我們將詳細分析如何透過異質性代理的微觀互動，構建出具備「蝴蝶效應」特徵的複雜適應系統（Complex Adaptive System），並探討如何利用動態知識圖譜與八卦協議來模擬資訊的不對稱性與衰減，從而實現真正意義上的「活體世界」 3。

## 2. 架構核心：黑板系統與知識編排的深度解析

在構建一個擁有成千上萬個智能代理（NPCs、派系、城鎮）的模擬世界時，首要面臨的挑戰是資訊的協調與通訊複雜度。如果每個代理都必須直接與其他所有代理進行通訊以獲取市場價格或政治狀態，系統的訊息傳遞複雜度將達到 $O(n^2)$，這在計算上是不可行的。因此，引入\*\*黑板架構（Blackboard Architecture）\*\*作為系統的中樞神經系統，成為了解耦資訊生產者與消費者的關鍵解決方案 4。

### 2.1 黑板隱喻在動態模擬中的角色重塑

黑板系統最初源自於早期的專家系統研究，其核心隱喻是一群專家圍繞著一塊黑板解決問題。在現代遊戲模擬中，黑板不僅僅是一個數據緩存區，它是一個主動的邏輯中介層。一個標準的黑板系統包含三個核心組件：**黑板（Blackboard）**、**知識源（Knowledge Sources, KS）以及控制組件（Control Component）** 5。

**黑板**是結構化的全域記憶體，存儲著當前的世界狀態（Solution Space）。例如，它記錄著「北方省份乾旱」、「鐵礦價格飆升」或「帝國向聯邦宣戰」等事實對象。這些數據並非靜態的數值，而是具備語義關聯的動態對象。**知識源**則是各個獨立的子系統或代理，它們具備特定的領域專長但不直接相互通訊。例如，「天氣系統」是一個知識源，它監測模擬時間並將「乾旱事件」寫入黑板；「市場系統」是另一個知識源，它讀取「乾旱事件」並調整糧食價格；「派系AI」則是第三個知識源，它讀取價格上漲的訊息並計算社會動盪指數。透過這種間接通訊，系統實現了高度的模組化與可擴展性——開發者可以隨時加入一個新的知識源（例如「瘟疫系統」），而無需重構現有的經濟或政治代碼 7。

**控制組件**則扮演著交響樂團指揮的角色。它負責監控黑板的狀態變化，並根據特定的啟發式規則（Heuristics）動態激活相應的知識源。在複雜的政治經濟模擬中，控制組件必須具備優先級調度能力。例如，當黑板上同時出現「糧食短缺」與「敵軍入侵」兩個事件時，控制組件應當優先調度「軍事反應模組」進行處理，因為生存威脅的效用權重高於經濟調整。這種事件驅動（Event-Driven）的架構確保了系統資源總是被分配給當前最緊迫的運算任務，從而提高了模擬的效率與響應速度 8。

### 2.2 層次化黑板架構與資訊過濾機制

對於一個涵蓋宏觀地緣政治與微觀個體生活的遊戲世界，單一的扁平化黑板將迅速成為效能瓶頸。為此，我們提出**層次化黑板架構（Hierarchical Blackboard Architecture）**，將數據與邏輯分層管理，以匹配不同尺度的模擬需求。

**表 1：層次化黑板架構的數據範疇與更新頻率**

| **層級 (Level)** | **作用範疇 (Scope)** | **數據類型 (Data Types)** | **更新頻率 (Frequency)** | **知識源範例 (KS Examples)** |
| --- | --- | --- | --- | --- |
| **大戰略層 (Grand Strategy)** | 全球/帝國 | 戰爭狀態、全球市場指數、科技樹進度、外交條約 | 低 (每日/每週) | 帝國AI、全球氣候系統、文明演化模組 |
| **運營層 (Operational)** | 區域/城市 | 區域庫存、犯罪率、瘟疫傳播、區域影響力圖 | 中 (每小時) | 城市管理者、商會AI、治安調度系統 |
| **戰術層 (Tactical)** | 隊伍/小區 | 局部路徑規劃、戰術掩體、目標選擇、局部噪音 | 高 (每幀/每秒) | 戰鬥AI、小隊協調器、物理碰撞系統 |
| **個體層 (Individual)** | 單一代理 | 生理需求（飢餓）、個人記憶、當前任務堆疊 | 高 (每秒) | NPC效用大腦、個人物品欄管理 |

這種分層結構不僅優化了性能，更重要的是實現了**資訊過濾（Information Filtering）與抽象化（Abstraction）**。一個處於「個體層」的農民代理（Peasant Agent）不需要知道「大戰略層」的全球鐵礦均價，他只需要存取「運營層」的本地市場麵粉價格。反之，帝國皇帝的AI不需要監控每一個農民的飢餓度，它只需要從「運營層」讀取匯總後的「區域動盪指數」（Regional Unrest Index）。區域黑板充當了資訊的聚合器（Aggregator），定期將微觀數據的統計特徵（如平均值、變異數）向上同步至全球黑板，而全球黑板則將重大狀態變更（如稅率調整、戰爭動員令）向下廣播。這種雙向的資訊流動機制模擬了真實世界中行政科層制的資訊傳遞特性，並自然地引入了資訊傳遞的時間延遲（Time Lag），為後續的政治博弈增加了策略深度 5。

### 2.3 基於動態知識圖譜的語義推理

為了支持更高級的政治演化與因果推理，黑板的數據結構應從傳統的關聯式資料庫升級為**動態知識圖譜（Dynamic Knowledge Graph, DKG）**。在知識圖譜中，實體（Entities）作為節點，關係（Relations）作為邊，構成了世界的語義網絡。

與靜態數據表不同，知識圖譜能夠直接表示複雜的社會關係與歷史脈絡。例如，一條數據不再僅僅是 Relationship(FactionA, FactionB) = -50，而是一個豐富的圖結構：(FactionA) --> (FactionB)。這種結構使得代理能夠進行**鏈式推理（Chain Reasoning）**。當代理需要決定是否信任某個陌生人時，它可以查詢圖譜：「我的盟友是否信任這個人？」或「這個人的所屬派系是否曾在過去背叛過我的家族？」。透過圖遍歷演算法（Graph Traversal Algorithms），AI可以發現隱藏的關係模式，預測潛在的聯盟破裂或新興威脅 11。

此外，動態知識圖譜支持**時序感知（Temporal Awareness）**。圖中的邊可以包含時間屬性，允許系統追溯關係的演變軌跡。這對於模擬「政治記憶」至關重要——一個派系可能已經與前敵國簽署了和平條約，但圖譜中殘留的「歷史仇恨」節點仍然會影響其信任度計算，這種歷史慣性（Historical Inertia）是政治演化模擬真實感的關鍵來源 13。透過將黑板系統構建在動態知識圖譜之上，我們賦予了遊戲世界「記憶」與「解釋」自身歷史的能力，為湧現敘事提供了肥沃的土壤。

## 3. 認知引擎：效用AI與有限理性決策

在黑板系統提供了數據基礎後，**效用AI（Utility AI）則賦予了代理處理這些數據並做出決策的能力。相較於有限狀態機（FSM）或行為樹（Behavior Trees），效用AI在模擬複雜經濟與社會行為方面具有顯著優勢，因為它能夠通過數學權重處理模糊性、權衡相互衝突的需求，並展現出更接近人類的有限理性（Bounded Rationality）** 1。

### 3.1 效用函數的數學構造與響應曲線設計

效用理論的核心假設是：代理在任何時刻都會選擇主觀期望效用（Expected Utility）最高的行為。一個行為 $A$ 的效用得分 $U(A)$ 通常由多個考量因素（Considerations）經過歸一化處理後加權合成。其通用數學形式可表示為：

$$U(A) = \left( \prod\_{i=1}^{n} C\_i(x\_i)^{w\_i} \right) \times M\_{feasibility}$$

其中，$C\_i(x\_i)$ 是輸入變量 $x\_i$ 的響應曲線函數，$w\_i$ 是該考量的權重，$M$ 是二元可行性掩碼（例如，若代理沒有足夠金幣，則購買行為的可行性為0，總效用歸零）。使用幾何平均數（連乘）而非算術平均數（連加）的設計至關重要，因為這確保了如果任何一個關鍵考量為零（例如「極度恐懼」或「無工具」），整個行為的效用將被否決，這符合邏輯直覺 1。

\*\*響應曲線（Response Curves）\*\*是賦予AI「性格」與「有機感」的靈魂。線性關係在現實決策中極為罕見，因此系統需廣泛採用非線性曲線 16：

* **Logistic曲線（S形曲線）：** 用於模擬生理需求（如飢餓、睡眠）。在需求較低時，效用增長緩慢；一旦超過某個閾值，效用呈指數級飆升，迫使代理放棄其他活動優先滿足生存需求。
* **指數衰減曲線（Exponential Decay）：** 用於模擬邊際效用遞減（Diminishing Returns）。例如，擁有的金幣越多，獲取額外金幣的效用就越低；或者對於「復仇」的渴望隨著時間推移而指數衰減 17。
* **拋物線（Parabolic Curves）：** 用於模擬「最佳區間」偏好。例如，稅率的設定——過低導致國庫空虛，過高導致民怨沸騰，AI統治者會尋求位於拋物線頂點的平衡稅率。

### 3.2 異質性代理與動態個性向量

為了展現豐富的社會湧現行為，代理必須具備**異質性（Heterogeneity）**。如果所有商人代理都使用相同的效用權重，市場將變得機械且易於預測。我們通過為每個代理分配一個\*\*個性向量（Personality Vector）\*\*來引入多樣性 18。個性向量包含了一組係數，用於修正基礎效用公式中的權重 $w\_i$。

例如，一個行為 Accept\_Bribe（接受賄賂）的效用計算可能包含 Monetary\_Gain（金錢收益）與 Risk\_of\_Caught（被捕風險）兩個考量。

* **貪婪型代理（Greedy Agent）：** 對 Monetary\_Gain 分配 $1.5\times$ 權重，對 Risk 分配 $0.5\times$ 權重。
* **謹慎型代理（Cautious Agent）：** 對 Monetary\_Gain 分配 $0.8\times$ 權重，對 Risk 分配 $2.0\times$ 權重。

更進一步，這些權重應當是**動態**的。透過\*\*強化學習（Reinforcement Learning）\*\*或簡單的經驗反饋機制，代理的個性會隨著「生活經歷」而改變。一個連續多次在貿易中被搶劫的商人，其 Risk\_Aversion（風險厭惡）係數會逐漸升高，導致他改變貿易路線或僱傭更多護衛。這種個體層面的適應性演化，是宏觀經濟結構變遷（如保鏢行業的興起）的微觀驅動力 20。

### 3.3 雙層大腦架構：生物本能與社會規範的衝突

為了模擬複雜的政治與社會行為，單一的效用層往往不足以表現「內心衝突」。我們提出**雙層效用架構（Dual-Layer Utility Architecture）**：

1. **生物/個人層（Biological/Personal Layer）：** 驅動飢餓、貪婪、恐懼等原始慾望。
2. **社會/派系層（Social/Factional Layer）：** 驅動榮譽、職責、法律、忠誠等社會性動機。

代理的最終決策是這兩層效用相互博弈的結果。例如，一名守衛發現小偷（是他的朋友）：

* *個人層效用：* Let\_Go（放行）具有高效用，因為 Friendship 權重高。
* 社會層效用： Arrest（逮捕）具有高效用，因為 Duty 與 Fear\_of\_Punishment 權重高。  
  系統計算兩者的加權總和。若社會層勝出，守衛會執行逮捕，但可能會產生「愧疚感」狀態，影響其與朋友的後續關係矩陣。若個人層勝出，守衛選擇放行，但這會增加其「腐敗度」（Corruption）變量。當大量代理的個人效用（如生存需求）長期壓倒社會效用（如法律遵守）時，宏觀上就會湧現出「社會失序」或「革命」的狀態，這正是政治演化模擬的精隨所在 20。

### 3.4 慣性與切換成本的數學建模

效用AI的一個常見陷阱是「頻繁抖動」（Dithering），即代理在兩個效用相近的行為間高頻切換。為了模擬真實的行為穩定性，必須引入慣性（Inertia）或遲滯（Hysteresis）機制。

正在執行的行為 $A\_{current}$ 會獲得一個額外的慣性獎勵（Momentum Bonus）。任何新行為 $B$ 的效用 $U(B)$ 必須顯著高於 $U(A\_{current}) + I\_{bonus}$ 才能觸發行為切換。

$$U\_{threshold} = U(A\_{current}) \times (1 + \beta\_{inertia})$$

其中 $\beta\_{inertia}$ 是慣性係數（例如 0.2，即新行為需好 20%）。這不僅解決了抖動問題，也模擬了人類行為的「承諾升級」（Escalation of Commitment）心理——一旦投入了一項任務（如戰爭或長途貿易），即使邊際效益下降，代理也傾向於堅持到底，直到局勢惡化到不可忽視的程度。這種機制為政治模擬中的「僵局」與「沉沒成本謬誤」提供了數學解釋 16。

## 4. 經濟模擬：基於代理的模型 (ABM) 與蝴蝶效應

經濟系統是驅動遊戲世界運轉的代謝引擎。在系統驅動型設計中，我們摒棄傳統的抽象經濟模型（如無限商店、固定價格），轉而採用**基於代理的計算經濟學（Agent-Based Computational Economics, ACE）**。這意味著每一個麵包的生產、運輸與消費，都是具體代理互動的結果 2。

### 4.1 生產函數與供應鏈的連鎖反應

經濟活動始於生產。代理（作為企業或家庭）根據生產函數（Production Function）進行資源轉換。一個典型的 Cobb-Douglas 生產函數適配於遊戲環境可簡化為：

$$Q = A \times L^{\alpha} \times K^{\beta} \times R^{\gamma}$$

其中 $Q$ 是產出（如武器），$L$ 是勞動力投入，$K$ 是資本投入（工具/工坊等級），$R$ 是原材料（鐵錠）。

代理會持續評估生產的邊際收益產品（Marginal Revenue Product, MRP）。如果黑板上的武器價格 $P\_{weapon}$ 上漲，使得 $P\_{weapon} \times Q > Cost\_{input}$，則鐵匠代理會增加工時或擴大生產規模。

這種微觀機制導致了供應鏈的**蝴蝶效應（Butterfly Effect）** 24。

1. **初始擾動：** 一場局部的小規模衝突摧毀了一座橋樑。
2. **一階效應：** 運輸煤炭的商隊被迫繞路，增加了運輸成本與時間，導致目的地城市的煤炭短缺。
3. **二階效應：** 由於缺乏煤炭，城市的冶煉廠停工，鐵錠產量暴跌，價格飆升。
4. **三階效應：** 依賴鐵錠的武器工坊無法生產劍和盾牌。
5. 系統性後果： 當鄰國發動入侵時，該城市因無法裝備新徵召的民兵而迅速淪陷。  
   在這個鏈條中，橋樑的破壞與城市的淪陷之間沒有直接的腳本連接，但透過ABM的供應鏈傳導，兩者建立了堅實的因果關係。為了強化這種效應，系統必須嚴格執行庫存衰減（Inventory Decay）與消費剛性。商品如果不會腐爛，囤積將緩衝所有衝擊；需求如果可以無限延後，短缺就失去了緊迫性。引入食物腐爛與裝備損耗機制，迫使經濟體系必須保持恆定的流動性（Velocity），從而放大了任何流動性阻斷的後果 26。

### 4.2 雙向拍賣機制與價格發現

為了實現真實的價格波動，市場交易不應基於固定價格，而應採用\*\*連續雙向拍賣（Continuous Double Auction, CDA）\*\*機制。這是現代金融市場的運作方式，也是模擬供需平衡最高效的算法。

* **訂單簿（Order Book）：** 每個市場節點（城市/交易站）維護各類商品的買單（Bids）與賣單（Asks）。
* **撮合邏輯：** 當 $Bid\_{price} \ge Ask\_{price}$ 時，交易達成，成交價通常取兩者中值。
* **價格發現（Price Discovery）：** 最後一筆成交價成為當前的「市場價格」。

透過CDA機制，價格能夠即時反映供需變化。例如，當戰爭爆發，大量代理發布「購買糧食」的買單（需求曲線右移），而賣單數量不變，撮合價格自然上漲。這種價格信號會透過黑板系統廣播，吸引遠處的**套利代理（Arbitrage Agents）**。這些商人代理的效用函數驅使他們在低價區買入、高價區賣出。他們的移動不僅平衡了各地的物價，更在空間上形成了肉眼可見的**貿易路線（Trade Routes）**。這些路線並非設計師繪製，而是由成千上萬次套利交易所湧現出的路徑積分 27。

### 4.3 貨幣供應與通貨膨脹控制

一個封閉的經濟系統必須嚴格管理貨幣供應量，否則將面臨通貨緊縮（流動性不足）或惡性通膨。

* **水龍頭與水槽（Faucets and Sinks）：** 遊戲通常需要貨幣生成源（怪物掉落、金礦）與回收源（稅收、修理費、消耗品）。
* **貨幣流通速度（Velocity of Money）：** 系統應監測貨幣的周轉率。當貿易頻繁時，同樣數量的金幣能支撐更大的經濟規模。
* **通膨指數（Inflation Index）：** 每個區域黑板應計算本地的通膨指數（一籃子商品價格的加權平均）。工資與任務獎勵應與該指數掛鉤（Indexation），以防止經濟崩潰。例如，如果麵包價格上漲10倍，但士兵的薪餉不變，士兵代理的效用函數將驅使他們大規模逃兵或叛變，這再次將經濟壓力轉化為政治危機。

## 5. 政治演化：賽局理論與派系動力學

政治是經濟的延續，也是資源分配衝突的最高形式。在本系統中，政治演化並非預設的劇情樹，而是派系之間以及派系內部基於利益博弈的動態平衡過程。

### 5.1 動態關係矩陣與指數衰減記憶

派系之間的關係不應是靜態的標籤（如「友好」、「敵對」），而應是一個受微分方程控制的動態變量。我們採用\*\*衰減-脈衝模型（Decay-Impulse Model）\*\*來模擬外交關係 $R(t)$ 17。

$$R(t) = R\_{baseline} + (R(t-1) - R\_{baseline}) \times e^{-\lambda \Delta t} + \sum \text{Events}$$

* **自然回歸（Regression to Mean）：** 關係傾向於隨著時間推移回歸到基線（$R\_{baseline}$）。這意味著如果不持續進行外交維護（貿易、贈禮），盟友關係會逐漸疏遠；同樣地，如果沒有新的衝突，仇恨也會隨時間淡化。
* **指數衰減係數（$\lambda$）：** 這是一個關鍵的個性參數，代表「遺忘速度」。
  + **寬容型領袖（Forgiving Leader）：** $\lambda$ 值較大，仇恨消退快。
  + **記仇型領袖（Vindictive Leader）：** $\lambda$ 值極小，甚至接近0，這意味著一次背叛將導致永久的敵對，模擬了歷史上長達百年的世仇 19。
* **事件脈衝（Impulse Events）：** 具體的互動（貿易協議、邊境摩擦、間諜活動）作為脈衝信號疊加在關係曲線上。這些事件的權重受雙方當前的實力對比與需求迫切度影響。

### 5.2 外交博弈中的重複囚徒困境 (IPD)

外交決策的核心可被建模為**重複囚徒困境（Iterated Prisoner's Dilemma, IPD）**。在每個回合（如每個月），派系需決定對鄰國採取「合作」（貿易/互不侵犯）或「背叛」（禁運/掠奪）的策略。

* **收益矩陣（Payoff Matrix）：**
  + 雙方合作：獲得貿易利潤（正和賽局）。
  + 雙方背叛：消耗軍費，雙輸（負和賽局）。
  + 單方背叛：背叛者獲得掠奪收益，受害者損失慘重。

為了實現逼真的政治演化，AI代理應採用演化策略而非固定腳本 21：

1. **以牙還牙（Tit-for-Tat）：** 初始合作，隨後複製對手上一輪的行動。這模擬了「榮譽」與「互惠」原則，容易形成穩定的長期聯盟。
2. **冷酷觸發（Grim Trigger）：** 合作直到對手背叛一次，之後永遠背叛。這對應於極端的不信任與鷹派外交。
3. **巴甫洛夫策略（Pavlov / Win-Stay, Lose-Shift）：** 如果上一輪結果好（收益高），則保持策略；如果結果差，則改變策略。這種策略能容忍偶爾的誤解，並具有機會主義特徵。

系統透過黑板記錄雙方的互動歷史 $H$，並計算對手合作的條件機率 $P(Cooperate|H)$。當內部資源 $Res$ 匱乏時，背叛的誘惑（掠奪收益的邊際效用）會急劇上升：

$$U(Defect) \propto \frac{Gain\_{loot}}{Res\_{current}}$$

這解釋了為什麼經濟崩潰往往導致極端民族主義與對外侵略——這是為了生存而進行的數學上的必然選擇（Desperation War）31。

### 5.3 派系內部的權力動力學：忠誠度與政變

派系並非鐵板一塊的單體（Monolith），而是由眾多子代理（貴族、將軍、公會領袖）組成的集合體。派系領袖的權力來自於對這些\*\*關鍵支持者（Key Supporters）\*\*的控制，這符合「獨裁者手冊」（Selectorate Theory）的邏輯。

* **忠誠度指標（Loyalty Metric）：** 每個關鍵支持者都有對領袖的忠誠度，取決於領袖是否滿足了他們的效用需求（如分配封地、減稅、賜予榮譽）。
* **權力權重（Power Weight）：** 不同支持者擁有不同的權力值（軍權、財權）。
* **政變閾值（Coup Threshold）：** 當 $\sum (Power\_i \times IsDisloyal\_i) > Threshold\_{coup}$ 時，系統觸發「政變」或「內戰」事件。

領袖AI必須在「外部擴張」與「內部收買」之間分配有限的資源。如果領袖將所有稅收用於對外戰爭而忽略了對國內貴族的賞賜，忠誠度將下降，導致前線大捷但後方政變的戲劇性結果。這種內部的不穩定性是政治模擬深度的重要來源 18。

### 5.4 程序化條約生成與約束滿足問題

為了超越預設的外交選項（如僅有的「結盟」、「宣戰」按鈕），系統應利用\*\*約束滿足問題（Constraint Satisfaction Problem, CSP）\*\*求解器來程序化地生成複雜條約。

* **輸入：** 派系A缺乏糧食但軍力強大；派系B糧食富餘但面臨蠻族威脅。
* **求解：** CSP算法搜索雙方效用空間的交集，生成提案：「派系B每月提供500單位糧食，換取派系A在B領土駐紮2個軍團並承諾防禦。」
* 評估： 雙方AI計算該提案的預期效用。若 $U(Proposal) > U(Status\\_Quo)$，條約簽署。  
  這種機制能產生歷史上真實存在的複雜外交安排，如附庸國、租借地、朝貢體系等，完全基於當下的供需與安全形勢 32。

## 6. 資訊動態：八卦協議與戰爭迷霧的數學實現

在真實世界中，全知（Omniscience）是不存在的。為了模擬真實的戰略深度，必須將資訊視為一種需要時間傳播、會衰減且會變異的物理實體。我們引入\*\*資訊擴散模型（Information Diffusion Models）\*\*來構建「戰爭迷霧」與「流言蜚語」。

### 6.1 八卦協議與傳染病算法

資訊在代理網絡中的傳播遵循八卦協議（Gossip Protocol），這在數學上與傳染病模型（Epidemic Algorithms）高度相似 33。

我們定義一個\*\*資訊包（Info-Packet）\*\*結構：

* Topic: "Death\_of\_King"
* Value: True
* Timestamp: $T\_0$
* SourceID: Agent\_123
* Confidence: $0.0 - 1.0$

當兩個代理在空間上接近（Spatial Proximity）或進行通訊時，他們執行**推-拉同步（Push-Pull Synchronization）**：

1. **Push：** 代理主動分享其認為具有高新聞價值（High Newsworthiness）的資訊包。
2. Pull： 代理詢問對方是否知曉其感興趣的話題。  
   由於頻寬限制（每次互動只能交換 $N$ 條訊息），代理必須根據效用函數對資訊進行排序。與「生存」相關的資訊（如「附近有強盜」）通常具有最高的傳播優先級。

### 6.2 資訊衰減與變異：SIRS 模型

資訊在傳播過程中會發生質變。我們採用修正的 **SIRS (Susceptible-Infected-Recovered-Susceptible)** 模型來模擬謠言的生命週期 35。

* **易感者 (S)：** 尚未聽說該消息的代理。
* **感染者 (I)：** 知道消息並積極傳播的代理。
* **恢復者 (R)：** 知道消息但認為其已過時或不再感興趣，停止傳播的代理。

變異機制（Mutation - The Telephone Game）：

每次資訊從代理A傳遞到代理B，都存在變異概率 $P\_{mut}$。變異受到代理屬性（如智力、誠實度）的影響 38。

* *原始事實：* 「領主生病了。」
* *第5次跳躍：* 「領主病危。」
* 第10次跳躍： 「領主已死，是公爵下的毒。」  
  這種\*\*語義漂移（Semantic Drift）\*\*創造了誤解與衝突。AI根據接收到的（可能是錯誤的）資訊做出決策，導致「悲劇性」的誤判。例如，一個忠誠的將軍可能因為聽信了關於國王背叛的謠言而發動「正義」的叛亂。

### 6.3 決策中的不對稱資訊與貝葉斯更新

代理的決策基於其主觀的\*\*認知世界模型（Perceived World Model）\*\*而非客觀真理。

* **信任度加權：** 代理對每個資訊來源都有信任評分。來自「宿敵」的資訊會被大幅降權（Skepticism）。
* 多源驗證（Triangulation）： 當代理收到相互衝突的資訊（來源A說「和平」，來源B說「開戰」）時，它使用貝葉斯推斷（Bayesian Inference）來更新後驗機率。若無法確信，代理可能會派遣「偵查員」去獲取第一手資訊，這本身就產生了一個新的行為任務（Quest）39。  
  這種資訊的不對稱性是戰略欺騙（Strategic Deception）的基礎。玩家或AI可以故意散布虛假情報（Disinformation），利用八卦網絡的傳播特性來誘導敵方主力誤判方向，從而實現「聲東擊西」的高級戰術湧現 40。

## 7. 空間與戰術動態：從地圖影響力到路徑攔截

經濟與政治的決策最終必須落實到空間上的移動與控制。

### 7.1 影響力圖與流場導航

對於宏觀戰略AI，我們將世界離散化為網格，並計算多層**影響力圖（Influence Maps）**。

* **軍事威脅圖：** 基於敵軍位置與投射範圍生成的勢場。
* **經濟價值圖：** 基於資源點與貿易線生成的吸引力場。
* **張力圖（Tension Map）：** 雙方勢力交疊的邊界區域。

AI的戰略移動可以視為在這些勢場中的**梯度下降（Gradient Descent）**：軍隊沿著威脅梯度的反方向移動以尋求安全，或沿著價值梯度移動以尋求佔領。流場（Flow Fields）技術允許成千上萬的單位在大規模地形上進行高效且協調的移動，而無需為每個單位單獨計算A\*路徑 41。

### 7.2 RRT 演算法在動態攔截中的應用

在微觀戰術層面，例如強盜攔截商隊或巡邏隊追捕間諜，傳統的A\*算法往往顯得僵硬且滯後。我們引入\*\*快速擴展隨機樹（Rapidly-exploring Random Trees, RRT）\**及其變體 RRT* 來處理連續空間中的動態攔截問題 42。

RRT算法通過在空間中隨機採樣並生長樹狀結構，能夠快速探索可行路徑。在攔截場景中，AI不僅考慮目標的*當前位置*，還對目標的*未來軌跡*進行採樣預測。

* **預測性採樣：** RRT樹向目標的預測位置生長。
* 動態避障： RRT能夠在移動的障礙物（如其他單位）之間即時重規劃路徑。  
  這使得AI表現出「預判」行為——它不會笨拙地跟在目標屁股後面跑，而是會嘗試切入內線或在前方埋伏（Ambush），展現出極高的戰術智能與逼真度 45。

## 8. 綜合：工程化湧現與反饋迴路調試

上述所有子系統（經濟、政治、資訊、空間）最終在黑板系統中交匯，形成一個緊密耦合的複雜系統。湧現行為的產生，依賴於系統內部\*\*反饋迴路（Feedback Loops）\*\*的設計與調試。

### 8.1 系統性反饋循環範例

讓我們追蹤一個完整的系統循環，展示各個模組如何協同工作以產生非預設的歷史事件：

1. **環境觸發（Environment）：** 隨機數生成器導致「南方省份」降雨量低於平均值（乾旱）。
2. **經濟響應（ABM）：** 南方農場產量下降。糧食庫存枯竭。黑板上「糧食」價格飆升。
3. **效用評估（Utility AI）：**
   * **農民：** Farm 的效用下降（無收成），Starve 的負效用上升。Revolt（叛亂）的效用因絕望而超過 Obey\_Law。
   * **商人：** Import\_Grain 的利潤預期暴漲。
4. **空間動態（RRT/Pathfinding）：** 商人派出商隊前往南方。同時，飢餓的農民轉變為「強盜」單位，使用RRT算法在道路上攔截商隊。
5. **賽局博弈（Game Theory）：** 商隊被搶。商人AI更新風險矩陣，決定停止南方貿易（Defect）。
6. **政治後果（Blackboard/Politics）：** 南方徹底斷糧。區域動盪指數突破閾值。地方貴族AI計算出中央政府無力救災，且自身忠誠度因賦稅壓力已降至低點，遂決定發動**政變**並宣佈獨立。
7. **資訊傳播（Gossip）：** 「南方獨立」的消息透過八卦網絡傳向帝國首都，但在傳播中變異為「南方勾結外敵入侵」。
8. **最終響應：** 皇帝AI基於錯誤的情報，判定這是一場外部戰爭而非內部飢荒，決定派遣大軍鎮壓而非賑災，導致局勢進一步惡化。

這個過程沒有任何一步是編劇寫好的，它是系統規則相互作用的自然結果。

### 8.2 穩定性與混沌的平衡

為了確保遊戲的可玩性，設計師必須在\*\*正反饋（Positive Feedback）**與**負反饋（Negative Feedback）\*\*之間取得平衡 24。

* **負反饋（穩定器）：** 當價格上漲，生產利潤增加，吸引更多生產者，導致供給增加，價格回落。這有助於維持經濟系統的長期穩定。
* 正反饋（驅動器）： 恐慌導致囤積，囤積導致更嚴重的短缺，引發更大的恐慌。或者，仇恨導致戰爭，戰爭製造新的暴行，加深仇恨。這有助於打破僵局，推動歷史進程與衝突爆發。  
  一個優秀的系統設計會利用\*\*阻尼係數（Damping Coefficients）\*\*來控制這些迴路。例如，設置庫存容量上限以限制囤積，或設置「戰爭疲勞度」（War Exhaustion）變量，隨著戰爭持續而指數級增加國內不滿，最終強行終止正反饋的戰爭循環，迫使和平（或崩潰）的到來。

## 9. 結論

構建一個能自行模擬經濟供需與政治演化的系統驅動型遊戲世界，不再是依賴海量的文本腳本，而是依賴精密的數學模型與架構設計。透過**黑板架構**實現模組間的數據解耦與協作，利用**效用AI**賦予代理人性化的決策權重，藉助**ABM**湧現出宏觀經濟現象，應用**賽局理論**推演外交動態，並通過**八卦協議**與**知識圖譜**構建充滿迷霧與變數的資訊環境，開發者可以創造出一個真正的「沙盒歷史引擎」。

在這個世界中，玩家不再是唯一的變數，而是巨大動態系統中的一個擾動源。玩家的行為會被系統吸收、放大或抵消，產生出連設計師都無法預料的蝴蝶效應。這正是系統性遊戲設計的魅力所在——它不講述一個故事，它孕育無限個故事。隨著硬體算力的提升與AI算法的優化，這種深度模擬的「活體世界」將成為未來高沉浸感遊戲體驗的黃金標準。

**表 2：關鍵技術與其解決的模擬問題對照表**

| **技術模組 (Technology)** | **解決的核心問題 (Core Problem Solved)** | **湧現的行為特徵 (Emergent Behavior)** |
| --- | --- | --- |
| **黑板系統 (Blackboard)** | 資訊協調複雜度 $O(n^2)$、模組解耦 | 全局事件響應、跨系統協同（如天氣影響經濟） |
| **效用AI (Utility AI)** | 機械化行為、缺乏優先級權衡 | 絕望行為、貪婪、機會主義、有機的生活節奏 |
| **ABM 經濟模型** | 靜態經濟、缺乏供需反饋 | 價格波動、職業變遷、供應鏈中斷的連鎖反應 |
| **重複賽局 (IPD)** | 外交隨機性、缺乏長期策略 | 聯盟形成、背叛、冷戰、信任的建立與崩塌 |
| **SIRS 資訊擴散** | 全知全能AI、缺乏情報戰略 | 謠言傳播、誤判、戰略欺騙、資訊滯後 |
| **RRT 路徑規劃** | 僵硬的追逐邏輯、戰術單一 | 埋伏、包抄、切入內線、利用地形的戰術機動 |

本報告所提出的架構藍圖，為實現這一願景提供了堅實的理論與工程基礎。

#### 引用的著作

1. Designing Multi-Agent Intelligence - Microsoft for Developers, 檢索日期：1月 8, 2026， <https://developer.microsoft.com/blog/designing-multi-agent-intelligence>
2. Agent-based computational economics - Wikipedia, 檢索日期：1月 8, 2026， <https://en.wikipedia.org/wiki/Agent-based_computational_economics>
3. ACE: A Completely Agent-Based Modeling Approach (Tesfatsion), 檢索日期：1月 8, 2026， <https://faculty.sites.iastate.edu/tesfatsi/archive/tesfatsi/ace.htm>
4. How does the blackboard learning system enhance AI training? - UMU, 檢索日期：1月 8, 2026， <https://m.umu.com/ask/q11122301573854565559>
5. Blackboard Systems - Stanford University, 檢索日期：1月 8, 2026， <http://i.stanford.edu/pub/cstr/reports/cs/tr/86/1123/CS-TR-86-1123.pdf>
6. Blackboard Architecture. Understanding and implementing… | by Platform sharks - Medium, 檢索日期：1月 8, 2026， <https://medium.com/@dtech9728/blackboard-architecture-47e1ab30357e>
7. Creating Decoupled Features: The Blackboard System - Game Developer, 檢索日期：1月 8, 2026， <https://www.gamedeveloper.com/design/creating-decoupled-features-the-blackboard-system>
8. Why AI-Powered Event-Driven Architectures Are Game-Changers Now? - Maruti Techlabs, 檢索日期：1月 8, 2026， <https://marutitech.com/ai-powered-event-driven-architecture/>
9. The Future of AI Agents is Event-Driven | by Sean Falconer | Medium, 檢索日期：1月 8, 2026， <https://seanfalconer.medium.com/the-future-of-ai-agents-is-event-driven-9e25124060d6>
10. Blackboard System in Ai - Lark, 檢索日期：1月 8, 2026， <https://www.larksuite.com/en_us/topics/ai-glossary/blackboard-system-in-ai>
11. getzep/graphiti: Build Real-Time Knowledge Graphs for AI Agents - GitHub, 檢索日期：1月 8, 2026， <https://github.com/getzep/graphiti>
12. Building AI Agents with Knowledge Graphs vs. Retrieval Augmented Generation - Medium, 檢索日期：1月 8, 2026， <https://medium.com/@senpubali7/building-ai-agents-with-knowledge-graphs-vs-retrieval-augmented-generation-a2730ec1915a>
13. The Role of Knowledge Graphs in Building Agentic AI Systems - ZBrain, 檢索日期：1月 8, 2026， <https://zbrain.ai/knowledge-graphs-for-agentic-ai/>
14. Predicting Natural Product-Drug Interactions with Knowledge Graph Embeddings - PMC, 檢索日期：1月 8, 2026， <https://pmc.ncbi.nlm.nih.gov/articles/PMC12150722/>
15. Heterogeneous Agent Modeling: Experimental Evidence - UC Irvine, 檢索日期：1月 8, 2026， <https://sites.socsci.uci.edu/~duffy/papers/HCE4_2018April.pdf>
16. Anyone have advice or general formulas for implementing an exponential or logarithmic growth curve in a game? (Additional details in comments). : r/gamedev - Reddit, 檢索日期：1月 8, 2026， <https://www.reddit.com/r/gamedev/comments/way42a/anyone_have_advice_or_general_formulas_for/>
17. Exponential decay - Wikipedia, 檢索日期：1月 8, 2026， <https://en.wikipedia.org/wiki/Exponential_decay>
18. ABIDES-Economist: Agent-Based Simulation of Economic Systems with Learning Agents, 檢索日期：1月 8, 2026， <https://arxiv.org/html/2402.09563v1>
19. AI personality - AoW4 Wiki, 檢索日期：1月 8, 2026， <https://aow4.paradoxwikis.com/AI_personality>
20. COOPERATIVE BEHAVIOR IN PRISONER'S DILEMMA GAME - Different models to capture the role of fairness, 檢索日期：1月 8, 2026， <https://tesi.luiss.it/12530/1/la_serra-costanza-tesi-2014.pdf>
21. Paying to Do Better: Games with Payments between Learning Agents - arXiv, 檢索日期：1月 8, 2026， <https://arxiv.org/pdf/2405.20880>
22. Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning - INF/PUC-Rio, 檢索日期：1月 8, 2026， <https://www.inf.puc-rio.br/wordpress/wp-content/uploads/2019/11/196293.pdf>
23. Advances in the agent-based modeling of economic and social behavior - PMC, 檢索日期：1月 8, 2026， <https://pmc.ncbi.nlm.nih.gov/articles/PMC8262124/>
24. Harnessing the power of agent-based models for mitigating supply chain risks and managing costs | AWS HPC Blog, 檢索日期：1月 8, 2026， <https://aws.amazon.com/blogs/hpc/harnessing-the-power-of-agent-based-models-for-mitigating-supply-chain-risks-and-managing-costs/>
25. Factory Shutdowns to TikTok Trends: How the Butterfly Effect Disrupts Supply Chains, 檢索日期：1月 8, 2026， <https://gainsystems.com/blog/how-the-butterfly-effect-disrupts-supply-chains/>
26. Dynamics 1 Differential Equations 2 Exponential Decay - Redwood Center for Theoretical Neuroscience, 檢索日期：1月 8, 2026， <https://redwood.berkeley.edu/wp-content/uploads/2018/09/dynamics.pdf>
27. From Agent Simulation to Social Simulator: A Comprehensive Review (Part 1) - arXiv, 檢索日期：1月 8, 2026， <https://arxiv.org/html/2510.18271v1>
28. How fast do we forget our past social interactions? Understanding memory retention with parametric decays in relational event models | Network Science, 檢索日期：1月 8, 2026， <https://www.cambridge.org/core/journals/network-science/article/how-fast-do-we-forget-our-past-social-interactions-understanding-memory-retention-with-parametric-decays-in-relational-event-models/A1D6F063D1DFBFE03E9660A9091996A1>
29. Evolving subjective utilities: Prisoner's dilemma game examples - Keio University, 檢索日期：1月 8, 2026， <https://keio.elsevierpure.com/en/publications/evolving-subjective-utilities-prisoners-dilemma-game-examples>
30. Prisoner's dilemma - Wikipedia, 檢索日期：1月 8, 2026， <https://en.wikipedia.org/wiki/Prisoner%27s_dilemma>
31. Resolving the iterated prisoner's dilemma: theory and reality - PubMed, 檢索日期：1月 8, 2026， <https://pubmed.ncbi.nlm.nih.gov/21599777/>
32. Revealed Preferences in a Sequential Prisoners' Dilemma: A Horse-Race Between Six Utility Functions - ifo Institut, 檢索日期：1月 8, 2026， <https://www.ifo.de/DocDL/cesifo1_wp6358_1.pdf>
33. Gossip Protocol: The Fascinating Algorithm That Powers Distributed Systems Through Rumor Spreading 🗣️ | by Tejas Nikhar, 檢索日期：1月 8, 2026， <https://tejastn10.medium.com/gossip-protocol-the-fascinating-algorithm-that-powers-distributed-systems-through-rumor-spreading-0a2e51e6985d>
34. Gossip protocol - Wikipedia, 檢索日期：1月 8, 2026， <https://en.wikipedia.org/wiki/Gossip_protocol>
35. Agent-Based Modeling of Rumor Propagation Using Expected Integrated Mean Squared Error Optimal Design - MDPI, 檢索日期：1月 8, 2026， <https://www.mdpi.com/2571-5577/3/4/48>
36. A multi-stage SIR model for rumor spreading - American Institute of Mathematical Sciences, 檢索日期：1月 8, 2026， <https://www.aimsciences.org/article/doi/10.3934/dcdsb.2020124>
37. SIRA: a model for propagation and rumor control with epidemic spreading and immunization for healthcare 5.0 - PMC - PubMed Central, 檢索日期：1月 8, 2026， <https://pmc.ncbi.nlm.nih.gov/articles/PMC9372983/>
38. (PDF) A rumor spreading model based on information entropy - ResearchGate, 檢索日期：1月 8, 2026， <https://www.researchgate.net/publication/319337426_A_rumor_spreading_model_based_on_information_entropy>
39. Information and Agreement in the Reputation Game Simulation - PMC - PubMed Central, 檢索日期：1月 8, 2026， <https://pmc.ncbi.nlm.nih.gov/articles/PMC9778237/>
40. Information-theoretic models of deception: Modelling cooperation and diffusion in populations exposed to "fake news" | PLOS One - Research journals, 檢索日期：1月 8, 2026， <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0207383>
41. Study of the Interception Scheme Based on A\* Path Finding Algorithm in Computer Game, 檢索日期：1月 8, 2026， <https://www.scirp.org/journal/paperinformation?paperid=101851>
42. Improved RRT Algorithm for AUV Target Search in Unknown 3D Environment - MDPI, 檢索日期：1月 8, 2026， <https://www.mdpi.com/2077-1312/10/6/826>
43. Rapidly exploring random tree - Wikipedia, 檢索日期：1月 8, 2026， <https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree>
44. Research on Unmanned Vehicle Path Planning Based on the Fusion of an Improved Rapidly Exploring Random Tree Algorithm and an Improved Dynamic Window Approach Algorithm - MDPI, 檢索日期：1月 8, 2026， <https://www.mdpi.com/2032-6653/15/7/292>
45. A Path Planning Model for Intercepting a Moving Target with Finite Obstacle Avoidance, 檢索日期：1月 8, 2026， <https://arxiv.org/html/2507.01447v1>
46. AI Projectile Intercept Formula for Gaming, without Trigonometry | by Andy Zou - Medium, 檢索日期：1月 8, 2026， <https://medium.com/andys-coding-blog/ai-projectile-intercept-formula-for-gaming-without-trigonometry-37b70ef5718b>