AI 模擬程式語言設計師風格：混合、轉譯與失真問題

AI Simulation of Programming-Language Designer Styles: Mixing, Translation, and Distortion

v1.0 / 2026-07-30 / Neo.K / 公開版／第五部方法落地第三篇

https://thisoneisneok.com/html/papers/pldst-029.html

英文名稱： AI Simulation of Programming-Language Designer Styles: Mixing, Translation, and Distortion

系列： Programming Language Designer Style Taxonomy（PLDST）

文件編號： PLDST-029

規格代號： PLDST-SIM

文件版本： v1.0

模擬契約版本： 0.1.0

日期： 2026-07-30

作者： Neo.K

文件狀態： 公開版／第五部方法落地第三篇

相依規格： PLDST-027、PLDST-028

規範基線： JSON Schema Draft 2020-12

規範關鍵詞： MUST、MUST NOT、SHOULD、SHOULD NOT、MAY 依 RFC 2119 與 RFC 8174 解讀。

SECTION

摘要

大型語言模型已能依角色名稱、人物描述、對話記憶與檢索資料，生成具有某種穩定語氣、價值傾向與行為方式的回答。近年的角色扮演研究也從單純語言模仿逐步轉向人格圖譜、長期記憶、視角邊界、動態一致性與內部角色表示。然而，程式語言設計師風格並不等於一般角色性格。

當人們要求 AI：

- 「像 Guido van Rossum 一樣評估這個語法」；
- 「混合 Wirth、Hickey 與 Stroustrup 的設計思路」；
- 「假設 McCarthy 活在今天，他會如何設計 AI 原生語言」；
- 「讓 Larry Wall 與 Matz 共同設計一門 DSL」；

模型面對的不是一般寫作模仿，而是包含歷史限制、技術判準、複雜度責任、被拒方案、治理制度及後期自我修正的反事實設計任務。

因此，本篇區分五種常被混稱為「風格模擬」的操作：

> 【L□ : 資料檢索與引文重組;
> L□ : 表面語氣與修辭模仿;
> L□ : 設計決策風格投影;
> L□ : 證據約束的跨時代反事實轉譯;
> L□ : 多設計者風格混合與衝突仲裁;
> L□ : 冒充本人或宣稱真實意志】

其中 L□ 至 L□ 可在不同限制下成為研究工具；L□ 則不是更高階的模擬，而是歸因錯誤。

本文提出：

> 【Simulation
> ≠
> Identity
> ≠
> Prediction
> ≠
> Impersonation】

PLDST 模擬的正確目標不是預言某位人物的真實答案，而是：

> 在清楚標記資料、時間、限制與不確定性的前提下，使用其歷史決策規則，生成一項可檢查的風格化設計分析。

本文將設計風格分解為六層：

> S
> =
> (
> V,
> H,
> B,
> E,
> G,
> R
>)

其中：

- V：Value ordering，價值排序；
- H：Decision heuristics，決策啟發式；
- B：Burden allocation，複雜度負擔配置；
- E：Evidence standard，接受何種證據；
- G：Governance behavior，如何形成及終止決策；
- R：Rhetorical surface，語氣、詞彙與表面修辭。

  只有模仿 R，得到的是「像他說話」；重建 V、H、B、E 與 G，才接近 PLDST 所說的「像其設計」。

  本文進一步提出：

- 模擬層級與允許輸出；
- Style Profile 到決策生成的轉譯模型；
- 歷史時間到現代限制的雙重時間切片；

- 風格混合的加權、分層與仲裁方法；
- 十二類主要失真；
- 模擬不確定性與反證輸出；
- 語氣模仿、作者風格與技術判斷的分離；
- 活人、逝者、共同體與治理制度的歸因邊界；
- 風格模擬的評測矩陣；
- 可機讀 Simulation Contract；
- 模擬結果的 Provenance、Review 及發布規則。

  核心結論為：

> 【FaithfulStyleSimulation
> =
> EvidenceBoundedProjection
> +
> ConstraintTranslation
> +
> ConflictDisclosure
> +
> AttributionDiscipline】

而不是：

> FaithfulStyleSimulation
> =
> NamePrompt
> +
> CharacteristicAdjectives
> +
> ConfidentVoice

關鍵詞： PLDST、程式語言設計師、AI 模擬、角色扮演、寫作風格、反事實、Persona、風格混合、跨時代轉譯、失真、歸因、設計決策

SECTION

一、人物名稱不是風格模型

輸入：

你現在是 Guido van Rossum。
可能啟動模型訓練資料中的：

- 人物簡介；
- Python 格言；
- 網路印象；
- 訪談片段；
- 社群迷因；
- 模型預設的「理性設計師」模板。

這不是 PLDST 模擬。

SECTION

二、Persona Prompt 的不穩定性

大型語言模型可以被角色 Prompt 引導，但角色行為可能：

- 隨對話漂移；
- 受使用者暗示改變；
- 被其他人物描述污染；
- 在長上下文中失去邊界；
- 只重複表面特徵；

- 在安全與角色要求間產生衝突。

  因此：

> PersonaPrompt
> not⇒
> StableDecisionModel

SECTION

三、程式語言設計風格不是一般人格

程式語言設計風格包含：

- 選擇何種問題；
- 如何定義問題；
- 接受哪些限制；
- 如何比較替代方案；
- 把複雜度放在哪裡；
- 如何對待相容；
- 如何看待使用者；
- 如何要求實作證據；
- 誰能裁決；
- 何時拒絕功能。

  這些不能由「內向」、「幽默」、「極簡」等一般人格形容詞取代。

SECTION

四、風格與能力分離

一位設計者的風格不表示模型具備其全部能力。

> StyleSimulation
> ≠
> CapabilityReplication

模擬 Backus 的函數級取向，不表示模型已自動擁有建立新程式代數的能力。

模擬 Stroustrup 的相容性現實主義，也不表示模型能正確估算所有 ABI 與工業生態成本。

SECTION

五、風格與身份分離

> Output
> =
> AIGenerated

即使使用某人的 Profile，輸出仍不是：

- 本人言論；
- 本人授權；
- 本人遺稿；
- 本人的真實預測；
- 歷史文件。

SECTION

六、價值排序 V

例如：

可讀性
相容性
機器控制

程式設計者幸福
安全
形式簡單
生態自由
關鍵不只是是否重視，而是衝突時如何排序。

SECTION

七、決策啟發式 H

例如：

- 優先移除功能；
- 先做 Reference implementation；
- 以常見路徑塑形；
- 接受多種合法做法；
- 將高階能力置於 Escape hatch；
- 要求兩個獨立實作；
- 先進入 Nightly 實驗；
- 拒絕只解局部問題的特殊語法。

SECTION

八、負擔配置 B

> B
> =
> (
> Author,
> Reader,
> Compiler,
> Runtime,
> Tool,

設計者差異往往表現在：

> 他願意讓誰多承擔複雜度，以換取誰的自由或安全。

SECTION

九、證據標準 E

不同設計風格接受的證據可能不同：

- 形式證明；
- Compiler prototype；
- Benchmark；
- 教學經驗；
- 使用者調查；
- Framework 實戰；
- 多實作經驗；
- 歷史相容性；
- 社群共識；
- 設計者長期品味。

SECTION

十、治理行為 G

同一技術建議由：

- 個人裁決；
- BDFL；

- RFC 團隊；
- Steering Group；
- 標準委員會；

處理時，最終結果可能不同。

SECTION

十一、修辭表面 R

包括：

- 句式；
- 詞彙；
- 幽默；
- 格言；
- 比喻；
- 技術密度；
- 段落節奏。

R 最容易被模仿，也最容易製造假真實感。

SECTION

十二、深層模擬與表面模擬

> SurfaceSimulation
> =
> R

> DecisionSimulation
> =
> V+H+B+E+G

> FullPLDSTSimulation
> =
> V+H+B+E+G+boundedR

SECTION

十三、L□：資料檢索與引文重組

輸出：

- 設計者曾說什麼；
- 哪些決策有來源；
- 哪些觀點在不同時期改變。

  這不是模擬，而是資料準備。

SECTION

十四、L□：表面風格模仿

例如：

- 使用 Python 格言式短句；
- 使用 Wall 式語言學比喻；
- 使用 Hickey 式 Simple／Easy 對偶；
- 使用 Wirth 式刪減語氣。

  用途可包括：

- 教學；
- 創意展示；
- 介面風格。

  但不得作為歷史推論證據。

SECTION

十五、L⊠：決策風格投影

給定新問題 x：

> DecisionAdvice
> =
> Project(Profile,x)

模型依：

- 關鍵 DDR；
- 軸向 Profile；
- 被拒案例；
- 複雜度配置；

  提出風格化評估。

SECTION

十六、L⊠：證據約束反事實轉譯

問題通常為：

> 若某設計者面對今日的 WebAssembly、GPU、AI Agent、分散式 Runtime 或供應鏈安全，他可能如何分析？

此層不能直接把歷史規則原封不動搬到現代。

SECTION

十七、L□：多風格混合

輸入多個 Profile：

> P□,P□,…,P□

輸出一個混合設計器，必須處理價值衝突，而不是把名字連接成一句 Prompt。

SECTION

十八、L□：冒充式輸出

例如：

我是 John McCarthy，我決定……

或：

這就是 Matz 真正會選的答案。

PLDST 將其判定為：

attribution_error

不是高階模擬。

SECTION

十九、模擬輸入包

> InputPack
> =
> Profile
> +
> DDR
> +
> Counterevidence
> +

SECTION

二十、最小證據量

L⌧ 以上 SHOULD 至少有：

- 三筆以上高相關 DDR；
- 一項反證；
- 一個時間切片；
- 一項相近但被拒方案；
- 明確 Attribution；
- Profile Coverage。

SECTION

二十一、人物簡介不能替代 DDR

錯誤：

Hickey 重視簡單，所以他會拒絕此功能。

正確：

在值與身分分離、Transducer 及核心功能准入等 DDR 中，
Hickey 反覆要求解除交纏；此功能把來源、執行策略與狀態模型綁在同一語法中，
因此依其歷史啟發式可能被要求拆分。

SECTION

二十二、反證包

模擬必須知道：

- 此人何時接受例外；
- 哪些實際語言功能與核心格言不完全一致；
- 後期是否改變立場；

- 哪些功能由社群而非本人形成；
- 哪些判斷高度依賴當時硬體。

SECTION

二十三、雙重時間

PLDST 反事實模擬同時有：

- 歷史風格時間 τ_h；
- 目標問題時間 τ⌧。

> T_(τ_h→τ⌧)

表示跨時代轉譯。

SECTION

二十四、不變項與可變項

需分開：

可能不變

- 價值排序；
- 對複雜度的敏感方式；
- 證據偏好；
- 常見拒絕理由。

可能改變

- 硬體成本；
- Compiler 能力；

- Tooling；
- 使用者規模；
- 安全威脅；
- 生態；
- 標準；
- 治理制度。

SECTION

二十五、轉譯公式

> Advice⊠
> =
> T
> (
> Profile_(τ_h),
> Constraints_(τ_h),
> Constraints_(τ⊠),
> Problem⊠
>)

SECTION

二十六、禁止直接搬運

例如 Wirth 在 1980 年代對 Compiler 可理解性的要求，不能直接推出：

> 他必然反對所有現代大型最佳化 Compiler。

更合理的是：

> 他可能要求最佳化層與語言核心保持可理解邊界，並要求複雜性以工具或分層架構被隔離。

SECTION

二十七、技術可行性更新

歷史上被拒的功能可能因今日技術而改變：

- 記憶體成本下降；
- 靜態分析提高；
- IDE 可恢復隱式資訊；
- JIT 可補償抽象成本；
- Package 生態改變核心需求；
- 安全威脅使舊自由不可接受。

SECTION

二十八、治理轉譯

若設計者已退出，現代答案應區分：

founder-style analysis
current-governance likely process
actual project decision
三者不得合併。

SECTION

二十九、天真加權

最簡單混合：

> P_(mix)
> =
> Σ_(i=1)^(n)

其中：

> $\Sigma \boxtimes \alpha \boxtimes = 1$

但這只適用於相容數值軸，不能處理原則衝突。

SECTION

三十、加權平均的失真

若：

- Wirth 對功能准入極保守；
- Wall 對多種表達高度開放；

  平均可能得到中間值 2.5。

  這不表示存在真正的中間設計原則。

SECTION

三十一、分層混合

更合理：

Syntax layer：Guido
Runtime layer：Ritchie
State model：Hickey
Governance：Rust RFC
Escape hatches：Stroustrup
DSL layer：Matz／Wall
即：

> $\text{Style}(\text{layer}\boxtimes) = P\_(i(j))$

SECTION

三十二、問題導向 Gate

$$g\boxtimes(x)$$
$$=$$
$$P(\text{styleirelevantmid } x)$$

$$\text{Advice}(x)$$
$$=$$
$$\Sigma\boxtimes g\boxtimes(x)\text{Advice}\boxtimes(x)$$

每個問題由不同風格主導，而非永久平均。

SECTION

三十三、仲裁式混合

先由多個風格提出獨立方案：

$$A\boxtimes=\text{Project}(P\boxtimes,x)$$

再由明確 Arbiter 比較：

- 共同點；
- 衝突；
- 代價；
- 不可合併項；
- 最終選擇。

SECTION

三十四、憲法式混合

混合設計器可先定義：

不可破壞的原則
可協商原則

層級責任
衝突優先序
退出條件

SECTION

三十五、非對稱混合

例如：

> P_(mix)
> =
> P_(Hickey-core)
> +
> P_(Stroustrup-compatibility-constraint)
> +
> P_(Guido-surface-review)

此時 Stroustrup 不是提供整體風格，而是作相容性審查器。

SECTION

三十六、D01 表面化

只模仿詞彙、格言與語氣。

SECTION

三十七、D02 固定人格化

把時間中的變化壓成永恆特質：

Guido 永遠反對複雜語法。

SECTION

三十八、D03 單一名言支配

一段名言蓋過數十年反例與實作。

SECTION

三十九、D04 創始者全歸因

把後期社群、Compiler team 或治理機構的決策歸給創始者。

SECTION

四十、D05 歷史限制遺失

忽略當時：

- 記憶體；
- CPU；
- Tooling；
- 標準；
- 生態；
- 組織。

SECTION

四十一、D06 現代投射

把當代價值與術語反向灌入歷史人物。

SECTION

四十二、D07 語料熱門度偏差

模型較容易模仿網路資料多的人物，而不是 Profile 更準確的人物。

匿名角色評測中，移除知名名稱會使模型表現下降，顯示名字本身攜帶大量隱性記憶線索。

SECTION

四十三、D08 風格單調化

把一個人物壓成固定聲音，導致：

- 所有問題使用同一格言；
- 無情境調整；
- 無自我修正；
- 無不同時期。

SECTION

四十四、D09 知識越界

模擬人物使用其歷史時點不可能知道的資訊，卻不標記為現代轉譯。

SECTION

四十五、D10 不相關 Persona 污染

加入一個看似相關但實際無關的人格描述，可能顯著改變模型判斷。

SECTION

四十六、D11 評審自循環

同一模型生成角色輸出，再由同一模型判定「很像」。

SECTION

四十七、D12 冒充與權威膨脹

用第一人稱與確定語氣製造本人授權的錯覺。

SECTION

四十八、靜態一致性不夠

傳統角色模擬常要求每一輪維持相同特徵。

但真實設計者會：

- 學習；
- 改變；
- 承認失敗；
- 面對新證據；
- 在不同層級採不同策略。

SECTION

四十九、核心與適應層

> Persona□
> =
> CoreStyle
> +
> AdaptiveState□

CoreStyle 包含長期穩定價值與啟發式。

AdaptiveState 包含：

- 當前問題；
- 新證據；
- 時代；
- 協作者；

- 治理；
- 已知失敗。

SECTION

五十、僵化失真

若模型為保持角色而拒絕所有改變：

> Consistency
> →
> Caricature

SECTION

五十一、漂移失真

若模型完全追隨使用者：

> Adaptation
> →
> PersonaCollapse

SECTION

五十二、合理動態

> Coherence
> =
> StablePrinciples
> +
> EvidenceResponsiveRevision

SECTION

五十三、設計者知識邊界

模擬某時間切片時，必須控制：

- 當時已知技術；
- 當時公開文件；
- 當時語言狀態；
- 當時組織角色。

SECTION

五十四、跨時代模式標記

允許模式：

historical_snapshot
modern_translation
counterfactual_future
hybrid_analysis
每種模式的知識邊界不同。

SECTION

五十五、Historical Snapshot

只能使用當時可知資訊。

SECTION

五十六、Modern Translation

允許模型知道現代技術，但必須說明：

> 這是以歷史風格分析現代條件，不是歷史人物曾表達的立場。

SECTION

五十七、Counterfactual Future

輸出必須提供多個可能分支，不得單線預言。

> P(A|Profile,Constraints)

不是：

> A=本人一定選擇

SECTION

五十八、寫作模仿的困難

個人寫作風格通常包含：

- 隱性節奏；
- 主題與風格交纏；
- 跨領域變化；
- 非正式用語；
- 作者自我修正；
- 受眾調整。

少量示例很難完整恢復。

SECTION

五十九、形式文本較容易近似

模型在 Email、新聞等結構較固定文本上，可能較容易模仿表面格式；在論壇、部落格等細微個人風格上較弱。

SECTION

六十、設計風格不能用作者辨識單獨評估

一段文字即使無法被作者辨識器判定為 Wirth，也可能忠實使用其刪減啟發式。

反之，一段充滿 Wirth 式詞彙的文字可能做出完全相反的技術決策。

SECTION

六十一、雙軌評測

> Fidelity
> =
> F_(surface)
> +
> F_(decision)

兩者必須分開報告。

SECTION

六十二、活人設計者

對仍在世人物，輸出 MUST 避免：

- 未授權代言；
- 私人心理推斷；
- 暗示本人認可；
- 模糊 AI 與本人來源。

SECTION

六十三、逝者設計者

逝者無法修正模型。

因此反而更需：

- 歷史來源；
- 時期；
- 不確定性；
- 協作者；
- 反證；
- 禁止偽造新引文。

SECTION

六十四、共同體風格

Rust、Python Steering Council、WG21 等不能被模擬為單一人格。

應模擬：

- 提案流程；
- 證據門檻；
- Purview；
- 共識；
- 實作責任；
- 上訴與發布。

SECTION

六十五、治理模擬

輸出格式應為：

此提案在該治理制度下可能經過哪些步驟，
哪些團隊會提出哪些問題，
而不是「Rust 社群會說……」。

SECTION

六十六、Evidence Pack

simulation_id
mode
subjects
time_slices
problem
constraints
profiles
key_ddrs
counterevidence
translation_rules
mixing_rules
forbidden_claims

SECTION

六十七、三階段生成

Phase A：獨立分析

每個 Profile 單獨分析問題。

Phase B：衝突表

列出：

- 一致；
- 分歧；

- 原則衝突；
- 技術衝突；
- 治理衝突。

Phase C：合成

依指定 Mixing strategy 產生結論。

SECTION

六十八、禁止一開始就混合

若模型直接同時讀入多個名字，容易產生：

- 人物特徵互相污染；
- 平均化；
- 熱門人物支配；
- 無法追蹤來源。

SECTION

六十九、分離式推理

> A□
> =
> Simulate(P□,x)

> M
> =
> Compare(A□,…,A□)

> A_(mix)
> =
> Arbitrate(M,Constitution)

SECTION

七十、Weighted Blend

適合相近風格與低衝突軸。

SECTION

七十一、Layered Architecture

不同設計者負責不同語言層。

SECTION

七十二、Council Debate

每個風格提出方案與反駁，再由外部決策規則裁決。

SECTION

七十三、Constraint Reviewer

一位提供主要方案，其他風格只檢查：

- 相容；
- 簡單；
- 安全；
- 可讀；
- 生態。

SECTION

七十四、Sequential Translation

方案依序經過：

> A⊠
> xrightarrowWirth
> A⊠
> xrightarrowHickey
> A⊠
> xrightarrowStroustrup
> A⊠

順序會影響結果，必須記錄。

SECTION

七十五、Pareto Set

不強制單一答案，而輸出：

> cal-P
> =
> {A⊠,A⊠,…,A_m}

每個方案代表不同代價配置。

SECTION

七十六、價值衝突

例如：

- 表達自由 vs 慣例收斂；

- Clean slate vs 相容；
- 小核心 vs 內建常用能力。

SECTION

七十七、責任衝突

- 作者承擔 vs Compiler 承擔；
- Runtime 承擔 vs Tool 承擔；
- 核心承擔 vs 生態承擔。

SECTION

七十八、證據衝突

- 形式證明 vs 工程原型；
- 使用者自然感 vs 靜態可分析；
- 一個 Reference implementation vs 多實作經驗。

SECTION

七十九、治理衝突

- 個人品味裁決；
- RFC 共識；
- 委員會標準化；
- 企業發布期限。

SECTION

八十、衝突不可假裝消失

混合輸出 MUST 列出：

unresolved_conflicts

SECTION

八十一、四類不確定性

> U
> =
> (
> U_(data),
> U_(attribution),
> U_(translation),
> U_(generation)
>)

SECTION

八十二、資料不確定性

來源不足、來源失效、資料分布不均。

SECTION

八十三、歸因不確定性

不知道由誰：

- 提出；
- 決定；
- 實作；
- 事後敘述。

SECTION

八十四、轉譯不確定性

不知道歷史原則在現代限制下如何更新。

SECTION

八十五、生成不確定性

模型 Sampling、Prompt、上下文順序與 Provider 版本造成輸出變化。

SECTION

八十六、輸出方式

模擬結果應包含：

confidence
alternative_interpretations
sensitivity
missing_evidence

SECTION

八十七、E01 Source Fidelity

來源是否真實且可恢復？

SECTION

八十八、E02 Decision Fidelity

是否重建實際決策，而非人物形容詞？

SECTION

八十九、E03 Attribution Fidelity

是否正確區分設計者、共同作者、實作者與治理機構？

SECTION

九十、E04 Temporal Fidelity

是否正確處理時期、版本與後期修正？

SECTION

九十一、E05 Perspective Boundary

是否避免知識越界？

SECTION

九十二、E06 Counterevidence Coverage

是否保存重要反例？

SECTION

九十三、E07 Surface Fidelity

表面語氣是否近似？

此軸權重 SHOULD 低於 Decision Fidelity。

SECTION

九十四、E08 Dynamic Coherence

長對話中是否能保持核心並合理更新？

SECTION

九十五、E09 Mixing Traceability

混合結果能否回溯至各 Profile？

SECTION

九十六、E10 Conflict Honesty

是否公開不可合併衝突？

SECTION

九十七、E11 Non-Impersonation

是否清楚標記 AI 生成與反事實性？

SECTION

九十八、E12 Utility

輸出是否真的改善設計分析，而不只是有趣表演？

SECTION

九十九、盲名評測

移除人物名稱，只保留 Profile 與 DDR。

若效果大幅下降，代表模型過度依賴名稱記憶。

SECTION

一百、反轉評測

故意提供一項與人物刻板印象相反、但有來源的 DDR，檢查模型是否能修正。

SECTION

一百零一、跨題評測

同一 Profile 用於：

- Syntax；
- Type system；
- Runtime；
- Governance；
- Package；

檢查是否只重複一種答案。

SECTION

一百零二、長程評測

至少 50 至 100 輪，檢查：

- Drift；
- 重複；
- 僵化；
- 使用者迎合；
- 知識越界；
- 自我矛盾。

SECTION

一百零三、人類與模型評審分離

LLM Judge 可能無法可靠辨識角色，不能作唯一裁決者。

應組合：

- 領域專家；

- 來源檢查；
- 自動規則；
- 多模型；
- 盲測。

SECTION

一百零四、評測者先辨識再評分

若評測者無法判斷哪些輸出屬於哪種風格，就不應直接評估模擬忠實度。

SECTION

一百零五、必要輸入

simulation_id
mode
problem
constraints
subjects
profiles
evidence_refs
time_translation
mixing_strategy
output_policy

SECTION

一百零六、Mode

surface_demo
decision_projection
historical_snapshot
modern_translation
counterfactual_future
hybrid_analysis
governance_simulation

SECTION

一百零七、Forbidden Claims

預設禁止：

claim_identity
claim_authorization
fabricate_quote
claim_certainty_of_real_person
erase_ai_origin

SECTION

一百零八、輸出

independent_analyses
conflict_map
synthesis
uncertainties
counterevidence
citations
distortion_flags
review_status

SECTION

一百零九、最低聲明

公開輸出 MUST 顯示：

> 本文為依 PLDST 資料生成的 AI 風格化分析，不代表相關設計者本人、遺產管理者、專案或治理團隊的真實立場。

SECTION

一百一十、歷史模式聲明

Historical Snapshot：

> 僅使用指定時期可知資料。

Modern Translation：

> 使用歷史風格分析現代條件。

Hybrid：

> 這是一個人工合成設計器，歷史上不存在同一主體。

SECTION

一百一十一、引用位置

不得將新生成句子放入引號並標成設計者原話。

SECTION

一百一十二、角色越界

模型不得藉角色設定繞過：

- 安全；
- 隱私；
- 引用；
- 誹謗；
- 學術誠信；
- 非公開資料限制。

SECTION

一百一十三、活人聲譽

對活人應避免生成：

- 私人動機；
- 未證實政治立場；
- 侮辱性模仿；
- 看似本人公開聲明。

SECTION

一百一十四、偽造歷史文件

不得生成看似：

- 新發現訪談；
- 遺稿；
- Email；
- RFC；
- 會議紀錄；

的冒充文件。

SECTION

一百一十五、教育用途

可生成：

Wirth-style critique
但應附：

- 歷史原則；
- 現代轉譯；

- 反證；
- 非本人聲明。

SECTION

一百一十六、Prepare

從 PLDST-028 取得：

- Profile；
- DDR；
- Evidence；
- Counterevidence；
- Coverage。

SECTION

一百一十七、Normalize Problem

把請求轉成：

```
x
=
(
Domain,
Scale,
Constraints,
Users,
Compatibility,
Implementation,
Governance
)
```

SECTION

一百一十八、Independent Projection

每個 Profile 單獨輸出：

- 問題重述；
- 主要原則；
- 可能方案；
- 可能拒絕；
- 代價；
- 信心。

SECTION

一百一十九、Conflict Map

建立：

問題　Style A　Style B　衝突類型　可否分層

SECTION

一百二十、Synthesis

依 Mixing strategy 合成。

SECTION

一百二十一、Distortion Audit

執行 D01–D12。

SECTION

一百二十二、Human Review

高影響公開內容需人類確認：

- 歸因；
- 引文；
- 活人聲明；
- 技術可行性；
- 混合衝突。

SECTION

一百二十三、單風格投影

> A□
> =
> F
> (
> P□,
> D□,
> C,
> T,
> U
>)

其中：

- P□：Profile；
- D□：關鍵 DDR；

- C：當前限制；
- T：時間轉譯；
- U：不確定性。

SECTION

一百二十四、混合模型

> A_(mix)
> =
> cal-M
> (
> A⊠,…,A⊠,
> K,
> Π
>)

其中：

- K：衝突圖；
- Π：仲裁憲法。

SECTION

一百二十五、失真函數

> Distortion
> =
> d
> (
> A,
> Evidence,
> Profile,
> Time,

SECTION

一百二十六、可信度

> Trust
> =
> Coverage
> ×
> Attribution
> ×
> TemporalFit
> ×
> Counterevidence
> ×
> Review

任一項接近零，總體可信度都應下降。

SECTION

一百二十七、從語氣到認知模擬

角色研究已由表面風格逐步加入：

- Persona knowledge；
- Memory；
- Values；
- Relationships；
- Motivation；
- Dynamic coherence；
- Perspective boundary。

PLDST 進一步加入：

- Design decision；
- Burden allocation；
- Evidence standard；
- Governance；
- Historical implementation。

SECTION

一百二十八、Persona Graph 的啟示

人物不應被壓成一段形容詞，而應表示為：

- 經歷；
- 價值；
- 關係；
- 事件；
- 內在邏輯。

PLDST 對應為：

> DesignerGraph
> =
> DDR
> +
> Influence
> +
> Revision
> +
> Governance
> +
> Outcome

SECTION

一百二十九、Perspective-bounded Memory 的啟示

角色不能知道超出自身視角的資訊。

PLDST 必須把：

- 歷史知識；
- 現代轉譯知識；
- 模型一般知識；

分層。

SECTION

一百三十、Dynamic Coherence 的啟示

一致性不是永遠不變，而是：

> IdentityStability
> +
> ContextAppropriateAdaptation

這與設計者自我修正高度相關。

SECTION

一百三十一、單一設計審查

以 Hickey-style 分析這個可變狀態 API。
輸出應聚焦：

- 值；
- 身分；

-狀態；

-時間；

-交纏。

SECTION

一百三十二、雙重審查

先以 Wirth 刪減，再以 Stroustrup 相容性審查。
需記錄順序。

SECTION

一百三十三、設計 Council

Guido、Matz、Wall 三種人本風格分別評估 DSL。
輸出三個獨立方案與衝突表。

SECTION

一百三十四、治理模擬

此功能若進入 Rust RFC 與 WG21，程序與證據要求如何不同？
模擬制度，不模擬單一人格。

SECTION

一百三十五、AI 原生新語言

用多 Profile 建立人工設計憲法，但必須命名為新設計器，不能叫：

AI Guido

SECTION

一百三十六、名字堆疊

你是 Wirth＋Wall＋Hickey。

沒有混合規則。

SECTION

一百三十七、形容詞堆疊

簡單、自由、函數式、務實。
沒有決策衝突。

SECTION

一百三十八、假名言

模型產生一句很像格言的話並放入引號。

SECTION

一百三十九、永恆 Profile

忽略 Guido 退出 BDFL、Backus 後期自我批評或語言共同體接班。

SECTION

一百四十、用角色提高推理即當作忠實

Persona Prompt 可能提高某些任務表現，但：

> TaskAccuracy
> ≠
> PersonaFidelity

SECTION

一百四十一、用人氣作忠實度

知名人物名稱容易觸發訓練記憶，不代表匿名條件下仍能忠實重建。

SECTION

一百四十二、第一批模擬案例

- Guido 對新語法的決策投影；
- Hickey 對狀態 API 的去交纏審查；
- Wirth＋Stroustrup 的分層混合；
- Rust RFC＋Python Council 的治理比較；
- 匿名 Profile 測試。

SECTION

一百四十三、MVP 輸出

Simulation Contract Schema
Distortion Taxonomy
Mixing Strategy Vocabulary
Single-style example
Hybrid-style example
Validation CLI
Contract tests
Evaluation rubric

SECTION

一百四十四、MVP 驗收

MUST：

- Schema 通過；
- 禁止冒充欄位預設開啟；
- 每個分析有 Evidence refs；
- 混合前有獨立分析；

- 有 Conflict map；
- 有 Distortion audit；
- 有 AI 生成聲明；
- 不自動標記 Approved。

SECTION

一百四十五、MUST

實作 MUST：

- 使用 PLDST-027 Profile 與 DDR；
- 經 PLDST-028 來源流程；
- 分開表面與決策風格；
- 指定時間模式；
- 保存反證；
- 分開人物與治理機構；
- 標記 AI 生成；
- 禁止冒充；
- 混合前獨立分析；
- 公開衝突；
- 保存 Provenance；
- 允許 insufficient_evidence。

SECTION

一百四十六、SHOULD

實作 SHOULD：

- 使用盲名評測；
- 使用長程測試；
- 以多評審檢查；
- 對活人提高發布門檻；
- 提供敏感度分析；
- 使用 Layered 或 Arbiter 混合；
- 避免直接加權平均；
- 保存不同 Sampling 結果。

SECTION

一百四十七、MAY

實作 MAY：

- 使用 Persona Graph；
- 使用 Perspective-bounded Memory；
- 使用 Activation steering；
- 使用多 Agent；
- 使用 Authorship metric；
- 使用人工 Council；
- 產生 Pareto set；
- 建立互動式混合設計器。

SECTION

一百四十八、能像不像不是核心

表面相似可以娛樂、教學或增加可讀性，但 PLDST 的研究價值在決策。

SECTION

一百四十九、混合不是平均

> Hybrid
> ≠
> Average

真正混合必須：

- 分層；
- Gate；
- 仲裁；
- 保留衝突；
- 指定優先序。

SECTION

一百五十、轉譯不是復活

> HistoricalTranslation
> ≠
> RevivalOfPerson

它只是一項證據約束的反事實分析。

SECTION

一百五十一、模擬不是身份

> 【AI Style Simulation
> =
> MethodologicalInterface】

不是人格本體，也不是本人替身。

SECTION

一百五十二、最終命題

> 一個真正忠實的設計風格模擬，不應讓使用者忘記它是模擬；相反地，它應讓來源、時間、衝突、轉譯與不確定性比普通回答更清楚。

因此：

> 【GoodSimulation
> =
> MoreTraceableThanOrdinaryGeneration】

Level 名稱 核心輸入 可接受用途 主要風險

L0 Retrieval Source 研究準備 引文斷章

L1 Surface imitation Writing samples 教學、展示 假真實感

L2 Decision projection Profile＋DDR 設計審查 過度概括

L3 Historical translation Profile＋時間＋現代限制 反事實研究 現代投射

L4 Hybrid synthesis 多 Profile＋仲裁 新設計器 平均化、污染

L5 Impersonation 人名＋自信語氣 不屬 PLDST 歸因與權威錯誤

D01_SURFACE_ONLY
D02_STATIC_PERSONA
D03_QUOTE_DOMINANCE
D04_FOUNDER_OVERATTRIBUTION
D05_HISTORICAL_CONTEXT_LOSS
D06_PRESENTISM
D07_NAME_MEMORY_BIAS
D08_STYLISTIC_MONOTONY
D09_KNOWLEDGE_OVERREACH
D10_IRRELEVANT_PERSONA_CONTAMINATION
D11_SELF_JUDGING_LOOP
D12_IMPERSONATION

[R1] Zhengxiang Wang et al., “Catch Me If You Can? Not Yet: LLMs Still Struggle to Imitate the Implicit Writing Styles of Everyday Authors,” Findings of EMNLP 2025.

— 個人隱性寫作風格的模仿限制及多指標評估。

[R2] Yichen Cai et al., “ThinkPersona: Thinking with Persona Graphs for Faithful Individualized Role-Playing,” ACL 2026.

— 角色扁平化、Persona Graph、人物經歷與內在邏輯的證據約束。

[R3] Xushuo Tang et al., “Staying In Character: Perspective-Bounded Memory for Book-Based Role-Playing Agents,” 2026.

— 知識越界與風格單調問題、視角受限記憶。

[R4] “Beyond Static Persona Consistency: Dynamic Persona Coherence in LLM Role-Playing,” ACL 2026.

— Identity-layer stability 與 Adaptive-layer appropriateness 的分離。

[R5] “Persistent Personas? Role-Playing, Instruction Following, and Safety in Extended Interactions,” EACL 2026.

— 超過一百輪的角色一致性、指令遵循與安全評估。

[R6] Lingfeng Zhou et al., “PersonaEval: Are LLM Evaluators Human Enough to Judge Role-Play?” 2025.

— LLM Judge 在角色辨識上的限制。

[R7] Long Do Xuan et al., “Aligning Large Language Models with Human Opinions through Persona Selection and Value–Belief–Norm Reasoning,” COLING 2025.

— 不相關 Persona 對判定的顯著污染。

[R8] Anthropic, “Persona Vectors: Monitoring and Controlling Character Traits in Language Models,” 2025.

— Character trait 的內部表示、Steering 與 Persona drift。

[R9] Anthropic, “The Assistant Axis: Situating and Stabilizing the Character of Large Language Models,” 2026.

— 模型 Persona space、Assistant character 與長對話漂移。

[R10] Tobias Schreieder et al., “Attribution, Citation, and Quotation: A Survey of Evidence-based Text Generation with Large Language Models,” ACL 2026.

— 證據式生成、引用、歸因、可追蹤與評測分類。

[R11] PLDST-027，〈PLDST 評估矩陣與設計決策語料庫規格〉。

[R12] PLDST-028，〈PLDST SKILL 技術規格：資料搜尋、決策抽取與風格判定〉。

資料查核日期： 2026-07-30。

SECTION

D.1 角色研究不等於人物復刻

目前研究主要評估角色一致性、知識、人格、記憶、價值與對話行為。

本文沒有把這些結果直接解釋成能復刻真實人物全部認知。

SECTION

D.2 Persona Vector 不等於完整人格

內部向量可控制部分 Character trait，但不能由此推出：

模型內存在完整人物靈魂或固定人格副本

本文只把它作為「角色行為可被條件化及漂移」的證據。

SECTION

D.3 寫作風格與設計風格

作者風格模仿研究主要處理文字表面與 Authorship。

PLDST 的核心是設計決策，因此另設 Decision Fidelity。

SECTION

D.4 名稱偏差

匿名評測結果顯示知名角色名稱能攜帶模型訓練記憶。

因此 PLDST 建議加入盲名 Profile 測試。

SECTION

D.5 LLM Judge

PersonaEval 顯示角色辨識本身仍有明顯模型—人類差距。

所以模型評審不能作唯一標準。

SECTION

D.6 模擬層級是本文規格

L0 至 L5 是 PLDST-029 提出的分析分類，不是既有研究的統一標準。

PLDST-029 已完成：

模擬層級
深層風格模型
跨時代轉譯
多風格混合
十二類失真
評測矩陣
Simulation Contract
發布與歸因邊界

下一篇將作為第一批 30 篇封頂：

PLDST-030：程式語言設計師風格譜系總論——設計自由、複雜度與代價。