# THE HYBRID FRANKENSTEIN: INTEGRATED SITEMAP

## 1. Theoretical Foundations of Void Management

The management of generative systems—whether they be the stochastic output of Large Language Models (LLMs), the emergent behavior of simulated agents, or the geometric permutations of digital construction toys—requires a fundamental re-evaluation of how we structure "possibility." We term this discipline **Void Management**. The "Void" is not merely empty space; it is the latent, high-dimensional state space of a generative system before output is collapsed into an artifact. In current paradigms, this Void is often treated as a resource to be maximized, a "wilderness" of infinite potential. However, the theoretical convergence of high-assurance software engineering, anthropological context analysis, and systems cybernetics suggests that the Void must instead be rigorously bounded. The "Hybrid Frankenstein" is the architectural synthesis of these disciplines—a system where **bounded structure enables unbounded possibility**.

### 1.1 Negative Space Programming: The Architecture of Constraint

The primary theoretical pillar of Void Management is **Negative Space Programming (NSP)**. Traditional software development, and indeed much of current prompt engineering, focuses on the "positive space"—defining what the system *should* do. NSP inverts this, asserting that the robustness of a generative system is defined by the rigorous specification of what it *cannot* do.

#### 1.1.1 The State Space and Correctness by Construction

A generative system operates within a **state space** that encompasses every possible permutation of its elemental units (tokens, voxels, logic gates). Within this infinite expanse lies a subset of **valid states**—configurations that adhere to the system's specification, invariants, and safety conditions. The remainder is the **negative space**—the realm of invalid, hallucinatory, or catastrophic states.1

In "Correctness by Construction," the goal is not to filter invalid states after generation (a "defensive" posture) but to render them **unrepresentable** within the system's grammar. If a state cannot be represented, it cannot be generated. This is achieved through the use of **Algebraic Data Types (ADTs)**, specifically **Product Types** (AND relationships) and **Sum Types** (OR relationships), which constrain the vocabulary of the system to valid combinations only.2

For example, in a "9×9 Grid" system (discussed later as the convergence implementation), an "invalid position" is not a coordinate that is checked and rejected; it is a coordinate that effectively does not exist in the system's type definition. This eliminates the "illegal state" at the architectural level, creating a "ghost evidence" of constraints—the deleted code and unwritten pathways that define the system's shape by their absence.2

#### 1.1.2 Mechanisms of Constraint: Contracts and Parsers

To enforce NSP in generative AI, we must move from "validation" to "parsing." The maxim "Parse, don't validate" 3 dictates that unstructured input (e.g., a user's natural language prompt) must be immediately transformed into a structured value that adheres to strict strictures.

* **Contracts:** These are the formal agreements between the "Director" (user) and the "Generator" (AI). They consist of **preconditions** (what must be true before generation), **postconditions** (what must be true after), and **invariants** (what must remain true throughout).
* **Parsers:** Instead of passing raw tokens into the Void, the system uses parsers to ensure that only inputs conforming to a specific **grammar** can enter the state space. This prevents the "Shotgun Parsing" anti-pattern, where checks are scattered throughout the pipeline, allowing "weird machines" to emerge from the gaps.3

#### 1.1.3 Security and Emergence: The Weird Machine

The failure to manage Negative Space results in the emergence of **Weird Machines**. In the context of **Language-Theoretic Security (LangSec)**, a Weird Machine is an unintended computational model that arises when a system processes input that violates its assumed grammar.4

In Generative AI, "hallucination" is simply the operation of a Weird Machine. The model, lacking a bounded grid, executes logic paths that exist in the negative space—creating "facts" that are syntactically correct but semantically void. By treating the context window as a formal memory structure rather than a scratchpad, Void Management neutralizes the Weird Machine. It treats "Context Rot" and "Lost in the Middle" phenomena 5 not as bugs, but as evidence of a Weird Machine operating on unmanaged state.

### 1.2 Context as Material: The Physics of Information

The second pillar treats "Context" not as an abstract concept, but as a physical material with distinct properties, economic costs, and degradation rates. This **Material Physics** of information governs the economics of the "Context Window."

#### 1.2.1 Material Physics: The Attention Budget

The **Context Window** is the "workspace" of the generative agent. However, it is governed by the **Attention Budget**. In Transformer architectures, the computational cost of attention scales quadratically ($O(n^2)$) with the number of tokens ($n$).5 This scarcity creates a physical limit on the "mass" of information the system can sustain.

* **Context Rot:** Just as biological material decays, context degrades. The "Lost in the Middle" phenomenon reveals that information placed in the center of a long context window is statistically less likely to be retrieved or attended to by the model.5 This "rot" necessitates active curation.
* **The KV Cache:** The **Key-Value (KV) Cache** is the physical instantiation of the model's memory.7 Void Management requires treating the KV Cache as a "writable" surface—a **material substrate** that must be sintered, compressed, and isolated to prevent the "pollution" of the attention budget by irrelevant tokens.

#### 1.2.2 Anthropological Foundations: Thick Output

The objective of managing this material is the production of **Thick Output**. Drawing on Clifford Geertz's "Thick Description," which distinguishes a physical "twitch" from a culturally significant "wink," Thick Output refers to generative responses that encompass deep cultural meanings, implicit hierarchies, and social nuances.8

Current AI models, operating on vast but shallow data, often produce "Thin Output"—surface-level correctness devoid of situated meaning. To achieve Thick Output, the context must be engineered to reflect **Situated Action** (Suchman), viewing the user not as a disembodied prompter but as an **agent** within a specific **environment**.8 The context window becomes the "field site" where **Actor-Network Theory** (Latour) is applied—mapping the relations between human actants and non-human artifacts (tokens, constraints).8

#### 1.2.3 Agentic Context Engineering

To manage the physics of context and produce Thick Output, we employ a tripartite agentic structure:

1. **The Generator:** Fills the Void with content.
2. **The Reflector:** Analyzes output against "Thick" criteria.
3. **The Curator:** Manages the KV Cache, executing pipeline operations of **Write**, **Select**, **Compress**, and **Isolate** to prevent context rot.5

### 1.3 Leverage Points: The Cybernetics of Control

Donella Meadows' "Leverage Points" framework provides the strategy for intervening in these complex systems. The shift from current generative approaches to Void Management represents a high-level intervention.

* **Level 1 (Paradigm):** The shift from **"AI Creates"** to **"AI Prepares."** The highest leverage point is recognizing that the human role is to define the boundaries (the grid), while the AI's role is to fill the possibilities within those boundaries.1
* **Level 2 (Goals):** Changing the system goal from "Maximize Output" (infinite generation) to "Bound Possibility" (constrained generation).
* **Level 3 (Rules):** Encoding "contracts" as rigid grid constraints.
* **Level 5 (Structure):** Designing **balancing loops** (constraints that dampen hallucination) rather than reinforcing loops (feedback cycles that amplify error).

## 2. Empirical Lineages: The Structural History

The theory of Void Management is not without precedent. It is the synthesized heritage of three distinct empirical lineages: the **Digital LEGO Ecosystem**, the **Modulex System**, and the **Simulation Art of Ian Cheng**. These histories serve as case studies in the tension between *fluid possibility* and *rigid structure*.

### 2.1 The Digital LEGO Ecosystem: A Study in Standard Survival

The digitization of the LEGO brick—a physical atom of creativity—offers a 30-year longitudinal study on how "standards" survive in the void of digital space.

#### 2.1.1 The Corporate Core: Failures of Proprietary Void

The LEGO Group's internal attempts to digitize the brick consistently failed because they prioritized proprietary control and "fluidity" over inspectable structure.

* **Panter (1986):** An early DOS-based tool for creating building instructions. It was effective but "lost" because it was tied to specific hardware and lacked an open file standard.10 It represents a "lost species" of void management.
* **SPU Darwin & L3D (1996-1999):** The "Strategic Product Unit Darwin" was a secretive, ambitious division tasked with creating "LEGO 3D" (L3D). Their vision was "fluid play"—a seamless transition between physical and digital.12 However, the project collapsed under its own weight ("technical debt") and the inability to "sinter" the digital assets into a usable standard. The **L3D** database was too heavy, too complex, and "rotted" before it could become a standard.12
* **LDD (LEGO Digital Designer):** A long-lived but ultimately dead-end tool. While it allowed for digital building, its proprietary format prevented true community integration, leading to its eventual "mothballing" in 2022.14

#### 2.1.2 The Community Core: Survival of the Textual

In contrast, the **Community Core** thrived by adopting **Negative Space Programming** principles implicitly.

* **LDraw (1995):** Created by James Jessiman, LDraw survived because it treated the brick as a **linguistic construct**. It defined a brick not as a binary object, but as a plain-text definition in a coordinate system.13 This **textual inspectability** meant that the "Void" was visible and manageable. Invalid states (broken geometries) could be debugged in text.
* **Void Parallel:** LDraw's success proves that **plain text coordinate systems** (grids) are superior to opaque binary blobs for long-term void management. The "open standard survival" is a lesson in **Legibility**.15

### 2.2 The Modulex System: The Metric Rationality

**Modulex**, a spin-off system from 1963, represents the "Ideal Grid" for Void Management. It differs from standard LEGO in its fundamental geometry, prioritizing "work" (planning) over "play."

#### 2.2.1 The Geometric Schism: 1:1 vs. 5:6

Standard LEGO bricks use a non-rational **5:6 aspect ratio** (height to width), derived from "play geometry" (making bricks easier for children to pull apart). Modulex, however, utilizes a perfect **1:1 cube** geometry based on a **5mm module** (the M20 system).16

* **Metric Rationality:** This 1:1 ratio made Modulex a **planning tool** rather than a toy. Architects used it to model buildings, and factory managers used it to plan production lines.17 The **Grid** was absolute.
* **Void Parallel:** In Void Management, we seek the "Modulex Rationality"—a coordinate system where every "token" (brick) has a precise, rational cost and position, unlike the "fuzzy" geometry of natural language.

#### 2.2.2 The Pivot: From Architecture to Planning to Signage

Modulex evolved from architectural modeling into a **planning board** system.16 This transition is crucial: it moved from *representing* reality (modeling a house) to *managing* time and resource voids (planning boards). The "9×9 Grid" concept in our theoretical framework is a direct descendant of the Modulex planning board—a physical instantiation of a **Constraint Satisfaction Problem (CSP)**.20

#### 2.2.3 The 2015 Suppression

In 2015, a revival of Modulex bricks was attempted. Molds were prepared, and test parts were produced. However, the LEGO Group (via KIRKBI) acquired the company and "buried" the project.17

* **Insight:** This "suppression" acts as a "Void Maintenance" operation. The Corporate Core could not tolerate a competing "standard" (a different grid) within its ecosystem. It enforced a **Singular Void** (System LEGO) by eliminating the "Weird Machine" (Modulex).

#### 2.2.4 The Pink Brick Problem

The **Pink Brick** serves as a recurring motif of "glitch" or "resistance" in this lineage.

* **In Video Games:** In *LEGO Marvel Super Heroes 2*, a "Pink Brick" is the object of a game-breaking glitch where characters cannot reach it, causing the game to crash.22 It represents an **unrepresentable state**—an object that exists in the void but cannot be "touched" (resolved) by the agent.
* **In Modulex:** The "Pink/Violet" colors were rare, short-lived parts of the palette, often associated with transition periods.24
* **Insight:** The Pink Brick symbolizes the **Negative Space**—the element that the system tries to purge (through patches or discontinuation) but which persists as "ghost evidence" of the system's boundaries.

### 2.3 Life After BOB: Simulation and the Director's Role

Ian Cheng's work provides the contemporary artistic lineage for **Agentic Context Engineering**.

#### 2.3.1 From Emissaries to BOB

Cheng's *Emissaries* trilogy (2015-2017) introduced "narrative agents" into open-ended Unity simulations. The central conflict was between the "story" (linear intent) and the "simulation" (chaotic emergence).26

* **BOB (Bag of Beliefs):** BOB (2019) is an AI creature composed of competing "demons" (sub-agents). BOB evolves based on interaction, managing its own "context window" of beliefs.27
* **Life After BOB:** This work introduces the "Narrative Cyborg." The "Director" sets the conditions (the Void/Grid), and the AI fills the details.27 This explicitly shifts the artist's role from "creating the frame" to "tuning the parameters" (Meadows' Level 9 leverage point).

#### 2.3.2 The Metis Suns Production Model

Cheng founded **Metis Suns** to formalize "Worlding"—the art of creating infinite games.29 The studio model here is **Director + Dev Team + Simulation**.

* **Worlding:** Cheng defines a World as a "gated garden" with laws and borders.30 It is the imposition of a **Grid** upon the "Wilderness" of Unity's physics engine. "Through meaningful constraints comes infinite possibilities".30

#### 2.3.3 Preservation Risks: The Unity Trap

Cheng's works face the **Unity Trap**—the risk that the proprietary engine (Unity) will become obsolete, rendering the "living" simulation dead (unrunnable).31

* **Preservation Strategy:** Emulation and "seamful" documentation are required. Just as LDraw survived because it was text, the "World" must be documented not just as visual output, but as **logic and constraints** (NSP) to survive context rot.31

## 3. The Void Itself

### 3.1 Defining the Void

The **Void** is the **Possibility Space** of the generative system. It is the mathematical set of all representable states.1

* **Inputs:** Possibility (Probability Distribution).
* **Outputs:** Artifacts (Sintered Scenes).
* **Distinction:** The Void contains *potential* facts, distinct from *actual* facts. Managing the Void is managing the collapse of potentiality into actuality.

### 3.2 The Scene and Context Rot

A **Scene** is the **Accumulated Context** at a specific moment—tokens, memories, and constraints held in the KV cache.5

* **Context Rot:** As the Scene expands, "attention" dilutes. The "Lost in the Middle" phenomenon dictates that without strict structure (grids), the center of the void becomes inaccessible.5
* **Sintering:** The process of applying "heat" (computational compute/intent) and "pressure" (constraints/NSP) to the loose "powder" of tokens to fuse them into a stable, usable artifact.3

## 4. Quasi-Creature & Agency

### 4.1 The Quasi-Creature

Generative systems like BOB or LLM Agents are **Quasi-Creatures**. They exhibit "fluent behavior" and "perceived agency" but lack somatic embodiment or stable grounding.27 They inhabit the **Uncanny Valley of Agency**: high perceived intelligence, low reliability.

### 4.2 Managing the Quasi-Creature

Void Management is the practice of **binding** the Quasi-Creature.

* **Human Agency:** The capacity to form *Intent*.
* **Machine Agency:** The capacity to execute *Action* within the Void.
* **The Conflict:** Without a "Grid" (NSP), the Quasi-Creature's actions drift into "Weird Machine" territory (hallucination). The Human Director must impose the "World" (laws) to constrain the Creature.4

## 5. Alignment & Legibility via Seamful Design

### 5.1 Process Alignment

Alignment is not just about the *what* (output) but the *how* (process). **Process Fidelity** requires that the system's internal reasoning matches the user's expectations.33

### 5.2 Seamful Design

Standard UI design strives for "seamlessness" (hiding the void). **Seamful Design** argues for exposing the "seams"—the boundaries, breakdowns, and transitions of the system.34

* **In Void Management:** The "9×9 Grid" is a **Seamful Interface**. It does not hide the constraints; it visualizes them. It shows the user *where* the context is rotting, *where* the boundaries of the simulation lie.35
* **Benefit:** This increases **Legibility**. When the user sees the "negative space" (what cannot be done), they trust the "positive space" (what is done).33

## 6. Interfaces & Tools: The 9×9 Grid

The research converges on a single architectural principle for implementation: the **9×9 Grid**. This is not merely a metaphor; it is the structural implementation of Void Management, derived from **Modulex Planning Boards** and **Sudoku CSPs**.

### 6.1 The 9×9 Grid Implementation

| **Feature** | **Description** | **Theoretical Basis** |
| --- | --- | --- |
| **Fixed Structure** | 81 cells maximum. | **NSP:** Limits state space. "Context Rot" is prevented by hard limits.5 |
| **Position as Type** | The location (Row/Col) defines the meaningful "type" of the content. | **Parsers:** "Structured Values" over loose tokens.3 |
| **Negative Space** | Invalid positions are physically unrepresentable. | **NSP:** "Illegal states unrepresentable".1 |
| **Visible State** | The entire grid is inspectable at a glance. | **Seamful Design:** Exposing the system state.35 |
| **Variable Instantiation** | Cell content varies; arrangement is the creative act. | **Worlding:** "Through meaningful constraints comes infinite possibilities".30 |

### 6.2 The Sudoku Analogy

The **Sudoku** puzzle is the perfect model for Generative Constraint Satisfaction.20

* **Variables:** 81 cells.
* **Domain:** Numbers 1-9 (or Tokens/Assets).
* **Constraints:** No repeats in Row/Col/Box.
* **Generative Act:** The AI (Solver) fills the grid not by "guessing" (hallucinating) but by **Backtracking Search**—exploring valid paths and pruning invalid ones.37
* **Thick Output:** A solved grid is a "Thick Output"—every cell is contextually related to every other cell through the rules of the system.

## 7. Convergence Point: The Unified Principle

The **Hybrid Frankenstein**—the integration of LEGO structure, AI generation, and Simulation dynamics—functions only when **Bounded Structure Enables Unbounded Possibility**.

### 7.1 The Unified Principle Table

| **Lineage / Theory** | **Principle** | **Convergence in 9×9 Grid** |
| --- | --- | --- |
| **NSP** | "Invalid states unrepresentable" | Grid eliminates invalid positions. |
| **Context as Material** | "Context has physics, costs money" | Grid bounds token expenditure (81 units). |
| **Leverage Points** | "Paradigm shift is highest intervention" | "AI Prepares" (Solver) replaces "AI Creates." |
| **Digital LEGO** | "Open standard survives 30 years" | Bounded structures (LDraw/Grid) endure. |
| **Modulex** | "Metric rationality for professional work" | Fixed grid enables precise collaboration. |
| **Life After BOB** | "Director sets conditions, emergence follows" | Human instantiates void, AI fills it. |

### 7.2 Conclusion

We must move beyond the illusion of "seamless" infinite generation. We must embrace the **Seamful Grid**. We must treat Context as **Material** that requires sintering. We must recognize that to create truly **Thick Outputs** (Geertz), we must first define the **Negative Space**—the silence that gives shape to the sound.

**Final Recommendation:** Implement the **9×9 Grid** as the standard interface for Agentic Context Engineering. Bound the possibility space to liberate the creative agent.

Citations Used:

1

#### Works cited

1. Exploring the Power of Negative Space Programming - Double Trouble, accessed December 10, 2025, <https://double-trouble.dev/post/negativ-space-programming/>
2. Negative Space Programming: What You Don't Write Matters Even More Than What You Do | by Samir Rustamov - Medium, accessed December 10, 2025, <https://medium.com/@samir_rustamov/negative-space-programming-what-you-dont-write-matters-even-more-than-what-you-do-fe28dd05648a>
3. Negative Space Programming: it's not bad, it's just misunderstood, accessed December 10, 2025, <https://www.loskutoff.com/blog/negative-space-is-misunderstood/>
4. Computing with Time: Microarchitectural Weird Machines - Communications of the ACM, accessed December 10, 2025, <https://cacm.acm.org/research-highlights/computing-with-time-microarchitectural-weird-machines/>
5. Effective context engineering for AI agents - Anthropic, accessed December 10, 2025, <https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>
6. Fighting Context Rot: The Essential Skill to Engineering Smarter AI Agents (According to Anthropic) - Inkeep, accessed December 10, 2025, <https://inkeep.com/blog/fighting-context-rot>
7. Arxiv今日论文| 2025-11-26 - 闲记算法, accessed December 10, 2025, <http://lonepatient.top/2025/11/26/arxiv_papers_2025-11-26>
8. "Too Much Alignment; Not Enough Culture": Re-balancing cultural alignment practices in LLMs - arXiv, accessed December 10, 2025, <https://arxiv.org/html/2509.26167v1>
9. Re-balancing cultural alignment practices in LLMs - arXiv, accessed December 10, 2025, <https://www.arxiv.org/pdf/2509.26167>
10. LEGO® building instructions through time | LEGO® History, accessed December 10, 2025, <https://www.lego.com/en-us/history/articles/d-lego-building-instructions-through-time>
11. The meaning or story behind the “reversed” Octan logo - lego - Reddit, accessed December 10, 2025, <https://www.reddit.com/r/lego/comments/1o23c6g/the_meaning_or_story_behind_the_reversed_octan/>
12. Inside the LEGO Group's Secretive Strategic Product Unit Darwin, accessed December 10, 2025, <https://www.lego.com/cdn/cs/set/assets/blt4212e2be20008c99/bits_n_bricks_s01e16_darwin_feature_and_transcript.pdf>
13. Inside the LEGO Group's Secretive Strategic Product Unit Darwin - Pad and Pixel, accessed December 10, 2025, <https://padandpixel.com/inside-the-lego-groups-secretive-strategic-product-unit-darwin/>
14. Inside one of the most important LEGO® games ever made, accessed December 10, 2025, <https://www.lego.com/cdn/cs/set/assets/blte53dbf634332aa73/bits_n_bricks_s04e43_feature_and_transcript.pdf>
15. Bits N' Bricks Season 5 Episode 47: The Rise of BrickLink Feature and Transcript - LEGO, accessed December 10, 2025, <https://www.lego.com/cdn/cs/set/assets/bltf643219fa5bd3d27/bits_n_bricks_s05e47_feature_and_transcript.pdf>
16. “Modulex? Is that real LEGO®? What is it?” - MiniBricks Madness, accessed December 10, 2025, <https://minibricksmadness.com/2010/09/12/modulex-is-that-real-lego%C2%AE-what-is-it/>
17. The end of the line for Modulex - Brickset, accessed December 10, 2025, <https://brickset.com/article/13806/the-end-of-the-line-for-modulex>
18. Modulex manuals, brochures and catalogues | New Elementary: LEGO® parts, sets and techniques, accessed December 10, 2025, <https://www.newelementary.com/2023/10/modulex-manuals-brochures-and-catalogues.html>
19. 5 MEP FP HZT - Memorial Memorial Town Hall Narrative 12-08-2016, accessed December 10, 2025, <https://selfservice.town.canton.ma.us/WebLink/DocView.aspx?id=62577&dbid=0&repo=CANTON-LASERFICHE>
20. Advanced Artificial Intelligence - Khalil Chebil, accessed December 10, 2025, <https://chebil.github.io/ai/>
21. Nanoblock – micro-size building bricks - theBrickBlogger.com, accessed December 10, 2025, <https://thebrickblogger.com/2014/09/nanoblock-micro-size-building-bricks/>
22. Lego marvel super heroes 2 glitch please help : r/legogaming - Reddit, accessed December 10, 2025, <https://www.reddit.com/r/legogaming/comments/p616ce/lego_marvel_super_heroes_2_glitch_please_help/>
23. GwenPool Mission - Oscorp Escapade :: LEGO® MARVEL Super Heroes 2 General Discussions - Steam Community, accessed December 10, 2025, <https://steamcommunity.com/app/647830/discussions/0/1489992713713764177/>
24. LEGO Colors | Rebrickable - Build with LEGO, accessed December 10, 2025, <https://rebrickable.com/colors/>
25. Modulex parts - Forums LDraw.org, accessed December 10, 2025, <https://forums.ldraw.org/thread-28856-post-57675.html>
26. Ian Cheng - LAS Art Foundation, accessed December 10, 2025, <https://www.las-art.foundation/explore/biographies/b608b7df-4829-4d24-93f4-bb2681ed2782>
27. faq - Ian Cheng, accessed December 10, 2025, <https://iancheng.com/faq>
28. Ian Cheng: BOB, accessed December 10, 2025, <https://bobs.ai/>
29. Ian Cheng - Enceladus Press, accessed December 10, 2025, <https://www.enceladus-press.com/ian-cheng>
30. Chimeric Worlding, accessed December 10, 2025, <https://chimeric-worlding.netlify.app/>
31. Technical narratives analysis, description and representation in the conservation of software-based art - Monoskop, accessed December 10, 2025, <https://monoskop.org/images/4/49/Ensom_Tom_Technical_Narratives_Analysis_Description_and_Representation_in_the_Conservation_of_Software-based_Art_2019.pdf>
32. Preserving Virtual Reality Artworks | Tate, accessed December 10, 2025, <https://www.tate.org.uk/documents/54/tate_pim_preservingvrartworks_01_00.pdf>
33. Design and Artificial Intelligence: Ethical, Plural, and Situated Perspectives - Revistas UDD, accessed December 10, 2025, <https://revistas.udd.cl/index.php/BDI/announcement/view/105>
34. Seamful AI for Creative Software Engineering: Use in Software Development Workflows, accessed December 10, 2025, <https://www.computer.org/csdl/magazine/so/2025/03/10857384/23VCaKhFy80>
35. Seamful XAI: Operationalizing Seamful Design in Explainable AI - arXiv, accessed December 10, 2025, <https://arxiv.org/html/2211.06753v2>
36. Solving Sudoku as a Constraint Satisfaction Problem (CSP) | by Yashkochar - Medium, accessed December 10, 2025, <https://medium.com/@yashkochar01/solving-sudoku-as-a-constraint-satisfaction-problem-csp-54cb553c3cab>
37. Constraint Satisfaction Problems (CSP) in Artificial Intelligence - GeeksforGeeks, accessed December 10, 2025, <https://www.geeksforgeeks.org/artificial-intelligence/constraint-satisfaction-problems-csp-in-artificial-intelligence/>
38. On Weird Machines, etc.. Thomas Dullien's (@halvarflake) recent… - Alex Gantman, accessed December 10, 2025, <https://againsthimself.medium.com/on-weird-machines-etc-2834b0913023>
39. Modulex Baseplate 100 x 250, Planning Board : Part MxBoard2 | BrickLink, accessed December 10, 2025, <https://www.bricklink.com/v2/catalog/catalogitem.page?P=MxBoard2>