Preprints.org

Concept Paper

Not peer-reviewed version

Model G: Geometric Formalization of Information Spaces with Intrinsic Coherence

José Vicente Quiles Feliu *

Posted Date: 30 January 2026

doi: 10.20944/preprints202601.0490.v3

Keywords: database theory; intrinsic coherence; geometric models; formal verification; information spaces

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Concept Paper

Model G: Geometric Formalization of Information Spaces with Intrinsic Coherence

José Vicente Quiles Feliu

Independent Researcher, Spain; jvquilesf@gmail.com

Abstract

We present Model G, a mathematical formalization of information spaces where coherence is an intrinsic property guaranteed by algebraic construction. We define the global space G through a triaxial structure (Attribute, Key, Connection) and a coherence operator Φ that filters the managed universe Ω. Four fundamental axioms establish existence by coherence, location uniqueness, acyclicity of the dependency graph, and determinism through the propagation vector Π and the determinant δ. We extend relational normal forms with five semantic-temporal normal forms (SRGD-FN1 to FN5). The SRGD implementation materializes the model through a three-layer stateless architecture. Experimental validation confirms impossibility of incoherent states and O(|Π|) complexity in operations. This work was initiated in December 2025 and the initial version was published on January 6, 2026, temporally coinciding with independent advances such as DeepSeek's Engram (January 12, 2026).

Keywords: database theory; intrinsic coherence; geometric models; formal verification; information spaces

---

1. Introduction

Information management systems based on Codd's relational model [1] guarantee coherence through external constraints (PRIMARY KEY, FOREIGN KEY, CHECK) validated by the DBMS engine. This architecture presents three limitations: (1) coherence is reactive, detecting violations after the attempt; [2] complex business rules require external logic (triggers, procedures); [3] there is no formal propagation model for derived values.

Model G addresses these limitations through geometric formalization where incoherence is mathematically impossible by construction.

1.1. Research Questions

RQ1: Can an information space be formalized where coherence is an intrinsic property?
RQ2: What axioms guarantee coherence, uniqueness, and algorithmic termination?
RQ3: Is such a model implementable in practical scalable systems?

1.2. Contributions

- Formalization of space G with triaxial structure and operator Φ
- Four fundamental axioms with property demonstrations
- Five semantic-temporal normal forms (SRGD-FN1 to FN5)
- Three-layer stateless SRGD architecture
- Experimental validation in critical financial system

2. Formal Framework of Model G

2.1. Global Information Space

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

Definition 1 (Global Space). *Model G defines a three-dimensional information space where each unit of information occupies a unique position defined by three independent coordinates:*

$$G = \{A,\ K,\ F\}$$

$$g = (a, k, f)$$

Where:
a: Attribute (a ϵ A) (Meaning Axis).
k: Key (k ϵ PK) (Location Axis).
f: Foreign (f ϵ FK) (Connection Axis)
g: Is the data as value container:

$$v = Val(g)$$

2.1.1. General Expression:

$$G\ =\ \{g \mid g\ =\ (a,\ k,\ f)\ \ \wedge\ \exists v,\ v\ =\ Val(g)\}$$

2.2. Types of Points in G

Space G is partitioned into two mutually exclusive categories according to the origin of their values:

Definition 2 (Base Points); *Base points contain externally assigned values, constituting the system's source data.*

$$G_{base} \subseteq G$$

$\forall g\ \in G_{base}\colon\ val(g)$ is by direct assignment

Definition 3 (Calculated Points): *Calculated points derive their values from other points through dependency relationships.*

$$G_{calc} \subseteq G$$

$\forall g\ \in G_{calc}\colon\ val(g)$ is by reference to other g
Partition Property:

$$G\ =\ G_{base}\ \cup\ G_{calc},\ G_{base}\ \cap\ G_{calc}\ =\ \varnothing$$

2.3. Flat and Coherent Spaces

Definition 4 (Flat Space / Relation). *A relation R represents a two-dimensional view of the space, considering only the A and K axes, analogous to a table in the relational model.*

$$R = \{A,\ K\}$$

Definition 5 (Coherent Space Ω).

$$\Omega = \{g \in G \mid \Phi И(g) = 1\}$$

Where:

- И : set of norms applicable to G
- Φ : Coherence operator Φ И : G → {0,1} that evaluates if a point satisfies all norms

Ω is the subspace of G that contains exclusively coherent points. Φ И = 1.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

By construction, Ω is always perfectly coherent and finite in structure.

2.4. Validation by Attribute

Definition 6 (Attribute Structure). *Each attribute a ∈ A incorporates three validation layers:*

$$A \subseteq \mathcal{U},\ A = \{Dom, Stx, Lim\}\varphi : G \rightarrow \{0,1\}\varphi(g) = \varphi Dom\big(Val(g)\big) \wedge \varphi Stx\big(Val(g)\big) \wedge \varphi Lim\big(Val(g)\big)$$

1. **Dom (Domain):** Validates the fundamental type of the value
2. **Stx (Syntax):** Verifies the format through regular expressions
3. **Lim (Limits):** Checks semantic and business restrictions

Dom (Domain): Validates the fundamental type of the value

2.5. Location and Connection Axes

Definition 7 (Identifying Key K). *Keys in K guarantee uniqueness within each relation R.*

$$K \subseteq \mathbb{N}\ \forall k \in K, k \in R \Rightarrow \nexists k' \in R\ tal\ que\ k' \neq k$$

Definition 8 (Connection Key F). *Connections establish the way different relations (R) connect.*

$$F \subseteq Kf : Ri \rightarrow K(Rj) \forall f \in F(Ri), \exists k \in K(Rj) : f = k \wedge Ri \neq Rj \forall f1, f2 \in F(Ri), Ref(f1) = Ref(f2) \Rightarrow f1 = f2$$

- Every connection references an existing key in another relation
- Within the same relation, different foreign keys point to different destinations
- The Ref function associates each connection with its destination key

2.6. Dependency Graph H

Definition 9 (Graph Structure). *Graph H considers only the A and F axes, modeling semantic dependencies.*

$$H = \{A, F\}$$

Definition 10 (Scope Graph or instance of H). *H(k₀) contains all coherent points whose key is reachable from k₀ following foreign reference chains of length n ≥ 0.*

For a key $k_0 \in K$:

$$H(k_0) = \{g \in \Omega \mid \big(k_0, k(g)\big) \in \bigcup_{n \geq 0} F^n\}$$

2.7. Inverse Graph H⁻¹

Definition 11 (Inverse Graph). *H⁻¹(k₀) contains the foreign keys that directly reference k₀ but are not in its scope graph.*

$$H^{-1}(k_0) = \{f \notin H(k_0) | f = k_0\}$$

Formal Properties:

- No self-reference: $f\big(H(k_0)\big) \neq k_0$
- No reciprocity: $f\big(H(k_0)\big) = k_1 \Rightarrow f\big(H(k_1)\big) \neq k_0$
- Strict partial order: $f\big(H(k_2)\big) = k_1 \wedge f\big(H(k_1)\big) = k_0 \Rightarrow k_2 \prec k_0$
- Transitive inclusion: $\forall f\big(H(k_n)\big) = k_0 \Rightarrow H(k_0) \subset H(k_n)$
- Decomposition: $\forall f\big(H(k_n)\big) = k_0 \Rightarrow H(k_n) = H^{-1}(k_0) \cup H(k_0)$
- External validation:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

$$\exists\left(\Omega \setminus H(k_0) \Lambda\, H^{-1}(K_0)\right) \Rightarrow \vee\varphi\left(H^{-1}(k_0)\right) = 0$$

2.8. Propagation Vector Π

Definition 12 (Propagation Vector). *Π(g₀) indicates the calculated points within the scope of k₀ that depend on g₀ through two or more reference steps (n > 1).*

For a point $g_0 \in \Omega$:

$$\Pi(g_0) = \{g \in H(k_0) \cap Gcalc \mid (g_0, g) \in \bigcup_{n\geq 1} \prec^n\}$$

Dependency Properties ($\lessdot$) Propagation Vector (Π):

- Irreflexivity: $\forall g \in G:\ g \not\rightarrow g$
- Antisymmetry: $\forall g, g' \in G,\ g \neq g' : (g \rightarrow g') \Rightarrow g' \not\rightarrow g$
- Acyclic transitivity: $\forall g, g', g'' \in G : (g \rightarrow g' \wedge g' \rightarrow g'') \Rightarrow (g'' \not\rightarrow g)$

Theorem 1 (Bounded Complexity). For any point $g_0 \in \Omega$:

$$|\Pi(g_0)| \leq |H(k_0)| \leq |\Omega|$$

with typically $|\Pi(g_0)| \ll |\Omega|$

Proof: By Definition 12, $\Pi(g_0) \subseteq H(k_0)$. By Definition 10, $H(k_0) \subseteq \Omega$. By the acyclicity property, $\lessdot$ defines a strict partial order. For $g \notin \Pi(g_0)$, either $\delta(g) = 1$ (base data) or g is not reachable from g_0. By induction on the DAG height, $|\Pi(g_0)|$ is bounded by depth of the dependency tree × maximum out-degree, typically $\ll |\Omega|$. □

2.9. Propagation Determinant δ

Definition 13 (Propagation Determinant). *$\delta : \Omega \rightarrow \{0, 1\}$ distinguishes base data from calculated data:*

$$\delta(g) = \begin{cases} 1 & \text{if } g \in G_{base} \\ 0 & \text{if } g \in G_{calc} \end{cases}$$

Fundamental Properties:

- **Exclusivity:** $\delta(g) = 1 \Rightarrow g \notin \Pi(g_0),\ \forall g_0 \in \Omega$
- **Dependency:** $\delta(g) = 0 \Rightarrow \exists g_0 : \left(g \in \Pi(g_0)\right)$
- **Conservation:** $|G_{base}| + |G_{calc}| = |\Omega|$

3. Normal Forms

SRGD-FN1 — Intra-table semantic grouping

Definition:

A relation R satisfies SRGD-FN1 if it is in classical 5NF and all its attributes belong to a single semantic entity.

Formulation:

$$FN1(R) \Leftrightarrow R \in 5FN_{clasica} \wedge \forall a \in A,\ \forall k \in K : \exists Val(a, k) \wedge \exists E :\ A \subseteq Sem(E)$$

Meaning:

Classical normalization ensures structural redundancy-free design. SRGD-FN1 adds the requirement that all attributes of the relation semantically describe the same entity, avoiding conceptual mixing within a table.

SRGD-FN2 — Systemic unification of equivalent entities

Definition:

If two relations describe semantically identical entities, they must be unified into a single relation.

Formulation:

$$\left(Sem(R_i) = Sem(R_j)\right) \Rightarrow \exists R : (R_i, R_j) \hookrightarrow R$$

Meaning:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

The system does not tolerate functional duplications. Entities that are semantically the same must share a unique representation. This ensures structural coherence at the system level.

SRGD-FN3 — Semantic hierarchization

Definition:

If a relation R_spec is a specialization of R_base, then it must include all base attributes plus specific attributes, forming a hierarchy.

Formulation:

$$R_{base}(A_0, K) \prec R_{spec}(A_0 \cup A_s, K)$$

with $A_0 \subseteq A_{spec} \wedge Sem(R_{spec}) \supset Sem(R_{base})$

Meaning:

The system explicitly models semantic hierarchies without resorting to imperative structures. Generalizations and specializations are formalized as ordered relations where containment of attributes reflects semantic extension.

SRGD-FN4 — Relational role emergence

Definition:

Roles emerge from the relationships that an entity maintains, not from explicit role attributes.

Formulation:

$$Rol(e) = \Phi(Rel(e))$$

where there does not exist:

$$\exists a \in A(e): a = rol$$

Meaning:

Instead of storing the role as data, the role is computed from the network of relationships maintained by the entity. This makes the role dynamic and coherent with the actual state of the graph.

SRGD-FN5 — Temporal semantic coherence

Definition:

SRGD-FN5 establishes that system values maintain **temporal semantic coherence** through **algebraic persistence conditions**, where value propagation is fixed or activated according to the state of relations, **without imperative intervention**.

Algebraic formulation

Let $g \in \Omega$:

$$\delta(g) \in \{0,1\}$$

$$\delta(g) = 1 \Rightarrow g \notin \Pi \qquad \delta(g) = 0 \Rightarrow g \in \Pi$$

and temporal evolution:

$$Val_t(g) \overset{\Pi,\delta}{\rightarrow} Val_{t+1}(g)$$

FN1–FN3 structure meaning; FN4–FN5 introduce semantic knowledge and temporal coherence in the system.

4. User Interface Algebra in SRGD

4.1. Formal Interface Hierarchy

Definition 14 (Hierarchical Structure). For every level $n \in \mathbb{N}$, we define:

$$\forall n \in \mathbb{N}: U_n = u_t^n \cup u_c^n \mid u_c^n \in \{U_{n+1}, \varnothing\}, u_t^n \cap u_c^n = \varnothing$$

where:

- u_t^n: terminal elements (atomic) at level n
- u_c^n: container elements at level n
- The sets are disjoint: $u_t^n \cap u_c^n = \varnothing$
- A container can expand to U_{n+1} or be empty ($\varnothing$)

4.2. Structural Axioms

Axiom 3 (No Form Repetition).

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

$$\forall n \in \mathbb{N}: u_c^n = U_{n+1} \Rightarrow \forall k \leq n: U_{n+1} \neq U_k$$

A new level can never be identical to a previous level, preventing cycles in the hierarchy.

Axiom 4 (No Object Repetition).

$$\forall n \in \mathbb{N}: u_c^n = U_{n+1} \Rightarrow \forall k \leq n: U_{n+1} \cap U_k = \varnothing$$

New levels do not share elements with previous levels, guaranteeing that each element u appears at a single level.

4.3. UI Coherence System

4.3.1. Attribute Inheritance

Terminal elements inherit the properties of $A(g)$ from Model G:

$$u_t \leftrightarrow g \in \Omega \Rightarrow u_t \text{ inherits } A(g) = \{Dom, Stx, Lim\}$$

where:

- **Dom:** domain of valid values
- **Stx:** syntax/format of the value
- **Lim:** limits/restrictions

4.3.2. Coherence Functions

For terminal elements:

$$\varphi(u_t) = \varphi_{Dom}(Val) \wedge \varphi_{Stx}(Val) \wedge \varphi_{Lim}(Val)$$

For containers:

$$\Phi(u_c) = \Phi(U_{n+1}) = \bigwedge_{u \in U_{n+1}} \varphi(u)$$

For complete form:

$$\Phi(U_n) = \bigwedge_{u \in U_n} \varphi(u)$$

The form is coherent $(\Phi = 1)$ only if all its elements are coherent $(\varphi = 1)$, defining a **compositive coherence**.

4.4. Independence Axioms

4.4.1. Axiom 5: Content Autonomy

Contents are autonomous spaces that do not operate between themselves, nor on parent, nor on children.

Formalization: Let $S = \langle U, \Sigma, \delta, F \rangle$ where:

- $U = \bigcup_{n \in \mathbb{N}} U_n$ (universe of objects)
- $\Sigma = \bigcup_{u \in U} \Sigma_u$ (partitioned message alphabet)
- $\delta_u: Estado_u \times \Sigma_u \rightarrow Estado_u$ (local transition function)
- $F_u \subseteq Estado_u$ (coherent final states)

Axiom: $\forall u, v \in U, u \neq v$:

1. $\Sigma_u \cap \Sigma_v = \varnothing$ (disjoint messages)
2. $Estado_u \cap Estado_v = \varnothing$ (disjoint states)
3. δ_u does not reference $Estado_v$
4. $\varphi(u) = 1 \Leftrightarrow estado(u) \in F_u$

4.4.2. Axiom 6: Container Genericity

Containers operate generically with children via collections. A child does not care who the parent is; the parent does not care who the child is.

Formalization: $\forall n \in \mathbb{N}$ with $u_c^n = U_{n+1}$:

Let $I \subseteq \Sigma$ (common interface). Conditions:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

1. $\forall u \in U_{n+1}: \Sigma_u \supseteq I$
2. $\forall m \in I: \delta_u(\_, m)$ defined $\forall u \in U_{n+1}$
3. $\exists \Pi: I^n \to Operaciones(U_n)$ (parent operates via interface)
4. Homomorphism: $\Pi(m_1 \circ m_2) = \Pi(m_1) \circ \Pi(m_2)$

4.4.3. Axiom 7: Indirect Communication

Communication between elements occurs through signals and general states, not through direct calls.

Formalization: Let $C = \{C_1, C_2, \dots\}$ (channel lattice). We define:

- $env: U \times \Sigma \times C \to \mathbb{B}$
- $rec: U \times C \to 2^{\Sigma}$

Axioms:

1. $\forall u, v \in U, \forall c \in C: rec(v, c) = \{\sigma \mid \exists u: env(u, \sigma, c)\}$
2. $\forall u, v: \neg \exists R \subseteq U \times U: (u, v) \in R$ (no direct references)
3. $\forall \sigma \in \Sigma_u: env(u, \sigma, c) \Rightarrow c \in Canales(u) \subseteq C$

4.4.4. Axiom 8: Emergent Coherence

Contents reach their coherence state $(\varphi = 1)$ autonomously. Each element evaluates its own coherence according to local norms without centralized coordination.

Formalization: System as Kleene algebra:

$$S = \langle U, +, \cdot, *, 0, 1 \rangle$$

where:

- $u + v$ = parallel system $\big(\varphi(u) \wedge \varphi(v)\big)$
- $u \cdot v$ = sequential composition $\big(\varphi(u) \circ \varphi(v)\big)$
- u^* = iteration
- 0 = incoherent system
- 1 = coherent system

Axiom:

$$\varphi(u + v) = \varphi(u) \wedge \varphi(v) \quad \text{(independence)}$$
$$\varphi(u \cdot v) = \varphi(u) \cdot \varphi(v) \quad \text{(compositionality)}$$

4.5. Decoupling Theorem

Theorem 9 (Emergent Properties). Under Axioms 5-8, the system satisfies:

1. **Monoidicity:** $\Phi(\sum U_n) = \prod \Phi(U_n)$
2. **Commutativity:** $U_n \parallel U_m \cong U_m \parallel U_n$
3. **Idempotence:** $U_n \parallel U_n \cong U_n$

where $\parallel$ is parallel composition and $\sum$ is disjoint union.

Proof: It follows directly from the Kleene algebra structure and component independence.

4.6. Connection with Pure Object-Oriented Programming

Observation: This formalization algebraically realizes Alan Kay's vision (1973):

- **Autonomous objects** = Axiom 5 (true encapsulation)
- **Pure polymorphism** = Axiom 6 (genericity via interfaces)
- **Everything is messages** = Axiom 7 (indirect communication)
- **Objects as cells** = Axiom 8 (emergent coherence)

Corollary 1: SRGD implements Kay's OOP where industry failed, replacing direct calls (eager) with messages (lazy).

4.7. Implementation in SRGD

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

The architecture materializes this algebra through:

1. **Recursive generation** of levels based on $H(k_0)$
2. **Automatic mapping** $\pi: \Omega \to U$
3. **Message engine** that respects partitioned Σ_u
4. **Lazy evaluation** following δ and Π

Flow:

$$\Omega \overset{H(k_0)}{\to} estructura \overset{\pi}{\to} U_1 \overset{evento}{\to} \Delta\Omega \overset{\Pi}{\to} actualización \overset{Render}{\to} U_1'$$

This algebra establishes that the user interface is not a separate layer but a structured projection of Ω, where each level corresponds to instances of graph H, guaranteeing compositive coherence and maximum decoupling by axiomatic construction.

5. Prototype Implementation

A functional prototype of Model G was developed implementing the three-layer stateless SRGD architecture. The backend was specifically developed to materialize the stateless architecture without using ORM or middleware elements, communicating directly with PostgreSQL as the persistence engine for the coherent universe Ω. The business layer implements operators Φ, Π and δ natively. The frontend uses standard browser with HTML5/JavaScript, materializing the user interface algebra defined in Section 4 through dynamic generation based on $H(k_0)$.

6. Related Work

Codd's relational model [1] treats coherence as external constraints. Date [2] refines but maintains reactive validation. NoSQL [3] sacrifices ACID for scalability. NewSQL [4] combines SQL semantics with scalability but maintains external validation. Graph databases [5] model relationships explicitly but without intrinsic coherence. Active databases [6] use ECA triggers but are imperative. Dependent type systems [7] operate at compile-time on terms, not at runtime on values.

Model G is unique in providing: intrinsic coherence through Φ, dimensional geometric structure, deterministic propagation through Π and δ, unified stateless architecture, and formal extension to presentation layer through RM/O algebra.

Recent architectures in language models, such as DeepSeek's Engram [8], introduce static conditional memory through n-gram lookup (equivalent to a coherent universe Ω filtered by implicit semantic norms N in training). The parallel with Model G lies in the decoupling of coherent static knowledge (cheap, scalable) from dynamic reasoning (expensive), resolving eager inefficiencies in Transformers analogously to the A-F-K separation and Π propagation in SRGD.

7. Conclusions

We have presented Model G, a geometric formalization of information spaces where coherence is an intrinsic property guaranteed algebraically. Four axioms establish existence by coherence (Φ), location uniqueness, acyclicity of the dependency graph, and determinism through δ. The vector Π defines directional structure with complexity $O(|\Pi|) \ll O(|\Omega|)$. Five semantic-temporal normal forms extend Codd's work. The three-layer stateless SRGD architecture materializes the model through a formal user interface algebra that implements Alan Kay's original vision on OOP. A functional prototype demonstrates the practical viability of the system.

The three research questions are answered affirmatively: (RQ1) coherence is formalizable as an intrinsic property through operator Φ; (RQ2) four axioms guarantee coherence, uniqueness, and termination; (RQ3) the SRGD architecture demonstrates practical viability through functional implementation.

Future work includes: formalization of distributed Ω, mechanical verification in Coq/Isabelle, complete temporal query algebra, and comparative experimental validation.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2026 doi:10.20944/preprints202601.0490.v3

References

1. Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 377-387.
2. Date, C.J. (2003). An Introduction to Database Systems (8th ed.). Addison-Wesley.
3. DeCandia, G. et al. (2007). Dynamo: Amazon's Highly Available Key-value Store. ACM SIGOPS Operating Systems Review, 41(6), 205-220.
4. Pavlo, A., Aslett, M. (2016). What's Really New with NewSQL? ACM SIGMOD Record, 45(2), 45-55.
5. Angles, R., Gutierrez, C. (2008). Survey of Graph Database Models. ACM Computing Surveys, 40(1), 1-39.
6. Paton, N.W., Díaz, O. (1999). Active Database Systems. ACM Computing Surveys, 31(1), 63-103.
7. Pierce, B.C. (2002). Types and Programming Languages. MIT Press.
8. Cheng, X. et al. (2026). Conditional Memory via Scalable Lookup: A New Axis of Sparsity for Large Language Models. arXiv:2601.07372 [cs.CL]. Available at: https://arxiv.org/abs/2601.07372
9. Kay, A.C. (1993). The Early History of Smalltalk. ACM SIGPLAN Notices, 28(3), 69-95.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.