

***DOMAINOLOGY: A Proposed General Science of Domains, Membership, Boundaries, and Reconfiguration*****Phan Thành Trung**

Independent Researcher, Vietnam

Citation: Phan Thành Trung (2026) DOMAINOLOGY: A Proposed General Science of Domains, Membership, Boundaries, and Reconfiguration. J. of Mod Phy & Quant Neuroscience 2(5), 1-14. WMJ/JPQN-199

Abstract

Domainology (Miền học) is proposed as a general research discipline for systems whose domains, membership relations, boundaries, and feasible couplings are themselves subject to change. Rather than treating a domain as fixed background, the framework makes membership a first-class state relation and separates identity preservation from external adaptation. It defines domain signatures, preservation contracts, Carry-II boundaries, compatibility and coverage, transdomain trajectories, bridge-domain synthesis, observation-dependent Dark Segments, and a canonical family of seal/open/plan/adapt/handover/validate/commit operators. Domain Membership Operators serve as a reference implementation. Prime-TB Cut and Prime Excalibur are incorporated as an optional factorized algebra for discretized boundary reconfiguration: unique factorization guarantees exact recovery on the integer grid, while no claim is made that nature is intrinsically prime-coded. D-Bench v1.0 provides 1,000 synthetic transition records across five scenarios and five methods. Adaptive paths solve obstacle cases that defeat straight handover; bridge synthesis solves a disconnected-feasibility case that path planning alone cannot; rate governance recovers fast transitions; and PrimeDMO reproduces the reconfiguration behavior with modest quantization overhead. The contribution is therefore a proposed field-level language and research program for studying how systems preserve identity while changing the domains to which they belong.

***Corresponding author:** Phan Thành Trung, Independent Researcher, Vietnam.

Submitted: 25.09.2026**Accepted:** 29.09.2026**Published:** 07.10.2026

Keywords: Domainology, Domain Membership Operator, Carry-II, Transdomain Dynamics, Domain Reconfiguration, Boundary Operatorics, Prime-TB Cut, Prime Excalibur, Observability, Synthetic Benchmark

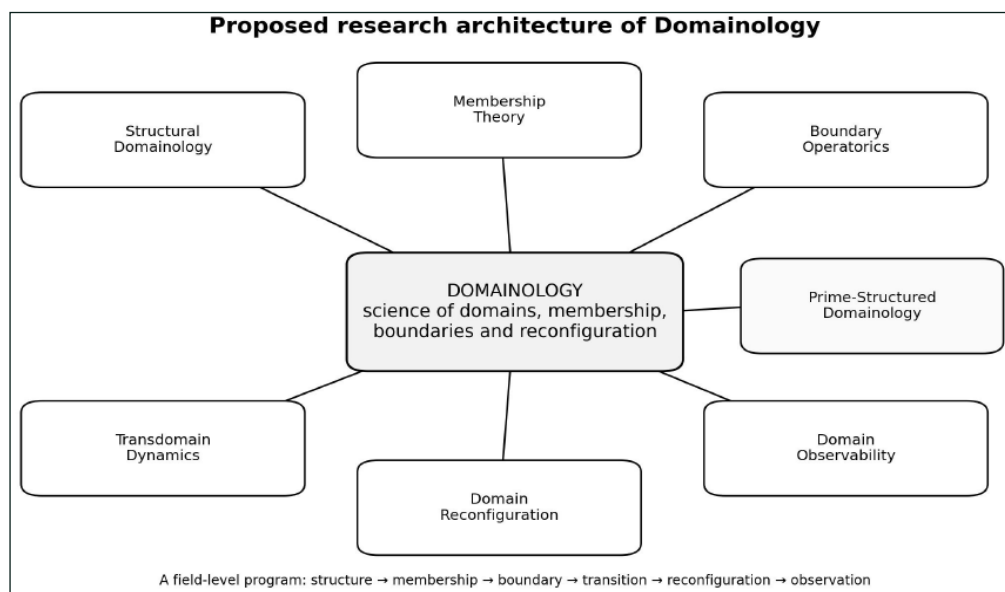


Figure 1: Proposed Research Architecture of Domainology.

Introduction

From Domain Assumptions to a Science of Domains

Many scientific and engineering models contain a hidden assumption: the domain in which a system is defined is fixed before the dynamics begin. A controller acts inside a plant model, a planner searches inside a configuration space, a software process runs inside a runtime environment, and a learning system is transferred from one data domain to another. The domain may change between experiments, but within the formal problem it is usually treated as context rather than as a first-class state relation.

Domainology is proposed here as a research discipline for making that relation explicit. It studies domains, domain membership, boundaries, compatibility, transitions, reconfiguration, observation, and composition. Its central question is not merely how a state changes, but how an object can remain identifiable while the regime that accepts, constrains, couples to, or observes that object changes. This shift turns domain membership from background metadata into an object of analysis and operation.

The proposal does not claim that every component is individually new. Viability theory studies admissible trajectories invariant-set methods formalize state constraints hybrid systems combine continuous and discrete dynamics observability separates latent states from measured outputs planning searches constrained spaces graph transformation studies-controlled rewrites and domain adaptation studies source–target mismatch. Domainology is instead a unifying operator-centered program: it treats the domain relation itself as mutable, validates identity preservation during that mutation, and explicitly distinguishes path finding from reconfiguration of the feasible relation [1-8]

The immediate precursor is the Domain Membership Operator (DMO), which formalizes a transition $\Omega_DM:(X, R_A) \rightarrow (X, R_B)$ while preserving a specified internal contract. The present paper generalizes that reference operator into a field-level architecture. DMO becomes one member of a broader family that includes boundary sealing and opening, compatibility handover, bridge-domain synthesis, observation operators, and representation-specific reconfiguration such as Prime-TB Cut and Prime Excalibur.

The resulting discipline is intentionally cross-domain. A 'domain' may be a physical interaction regime, a computational environment, a control mode, a communication protocol, a model class, a data-generating regime, or an abstract dynamical context. Domainology does not require these examples to be physically equivalent.

It asks which mathematical roles they share and which operator contracts must be satisfied before a transition can be called valid.

Definition and Scope of Domainology

Definition 1 (Domainology). Domainology is the systematic study of domains and the relations by which objects, states, processes, or models become members of them; of the boundaries that preserve or alter those relations; and of the operators that govern transition, synthesis, observation, and reconfiguration across domains.

The proposed field is organized around eight primitive questions: What constitutes a domain? What makes an object a member of it? Which properties define the identity of the object rather than the domain? Which boundary variables may change without destroying that identity? When is a direct transition feasible? When must an intermediate or bridge domain be synthesized? What can observers infer when visibility is domain-dependent? How do multiple domain transformations compose? These questions are broad enough to support a research program but narrow enough to admit explicit state variables and falsifiable algorithms.

A useful slogan is: identity belongs to the persisting object or trajectory; membership belongs to the relation between that object and a domain. This separation prevents a common category error in which changing environment is confused with destroying the object, or preserving the object is confused with freezing every microstate. The preservation contract can be coarse-grained, structural, functional, topological, informational, or physical depending on the application.

The term 'transdomain' is used operationally. It denotes a change in modeled domain membership. It does not, by itself, mean travel between universes, teleportation, or any extraordinary physical process. A containerized program migrated between runtimes, a controller switched between plant regimes, and a model transferred between data distributions are all legitimate transdomain examples in this formal sense.

Table 1: Foundational Primitives of Domainology

Primitive	Meaning
Object X	Persisting entity, process, model, or state carrier.
Domain R	Regime of admissible states, couplings, dynamics, observations, and constraints.
Signature Σ_R	Operational embedding of a domain into a comparison space.
Preservation contract I_X	Properties/relations that define identity at the chosen scale.
Boundary Π_X	Interface separating protected internal relations from adaptable external coupling.
Compatibility q_R	Raw fit between the current boundary configuration and a candidate domain.
Membership μ_R	Normalized relative allocation across candidate domains.
Coverage C_X	Best raw compatibility; detects null-membership situations.
Trajectory $\gamma(t)$	Planned evolution in domain-signature space.
Validator V	Test's identity, mismatch, coverage, resources, and application constraints.

Foundational Objects and Formal Language

A Domainology model begins with an object X and a domain R . A domain is represented as $R = (S_R, C_R, G_R, O_R, Q_R)$, where S_R is an admissible state set, C_R a coupling or interaction specification, G_R an evolution rule or generator, O_R an observation map, and Q_R a collection of constraints or invariants. A feature map Φ embeds the operationally relevant properties of R into a domain-signature space M_Σ , giving $\Sigma_R = \Phi(S_R, C_R, G_R, O_R, Q_R)$.

The object carries a preservation contract $I_X = (I_X, R_X, Q_X, D_X)$. The symbols denote identity descriptors, protected internal relations, required conserved quantities or constraints, and admissible internal dynamics. The contract defines an equivalence class of states that count as the same persisting object at the chosen modeling scale. If $e_I(t) = d_I(I_X(t), I_X(0))$, a valid transition requires $e_I(t) \leq \varepsilon_I$ throughout the protected interval.

A boundary Π_X separates protected internal structure from adaptable external coupling. Its abstract form is $\Pi_X = (\partial\Pi^-, \partial\Pi^+, I_X, b(t), V)$, where the inner boundary $\partial\Pi^-$ enforces the preservation contract and the outer boundary $\partial\Pi^+$ exposes a configurable signature $b(t)$ to the current domain. This is a functional distinction, not a claim about a literal material shell. A boundary can be software, a controller interface, an impedance-matching layer, a protocol adapter, a mathematical constraint, or, in a physical specialization, a measurable interface.

Raw compatibility with a candidate domain is defined by $q_R(X, \Pi) = \exp[-\beta d_\Sigma(b, \Sigma_R)]$. Normalized membership $\mu_R = q_R / \sum_j q_{Rj}$ is useful for relative assignment, but it cannot detect the case in which all raw compatibilities are simultaneously poor. Domainology therefore keeps an independent coverage variable $C_X = \max_R q_R$. A membership transition is covered only when C_X remains above a specified minimum $q_{\min}$.

A domain-membership path $\gamma: [0, T] \rightarrow M_\Sigma$ specifies the desired trajectory of external compatibility. Boundary mismatch is $e_\Sigma(t) = d_\Sigma(b(t), \gamma(t))$. A minimal valid transition therefore has two simultaneous obligations: preserve the object's internal contract and keep the external boundary close enough to a feasible compatibility trajectory. This dual accounting is one of the main reasons to distinguish Domainology from ordinary geometric motion.

$$R = (S_R, C_R, G_R, O_R, Q_R)$$

$$\Sigma_R = \Phi(S_R, C_R, G_R, O_R, Q_R) \in M_\Sigma$$

$$q_R(X, \Pi) = \exp[-\beta d_\Sigma(b, \Sigma_R)], \mu_R = q_R / \sum_j q_{Rj}, C_X = \max_R q_R$$

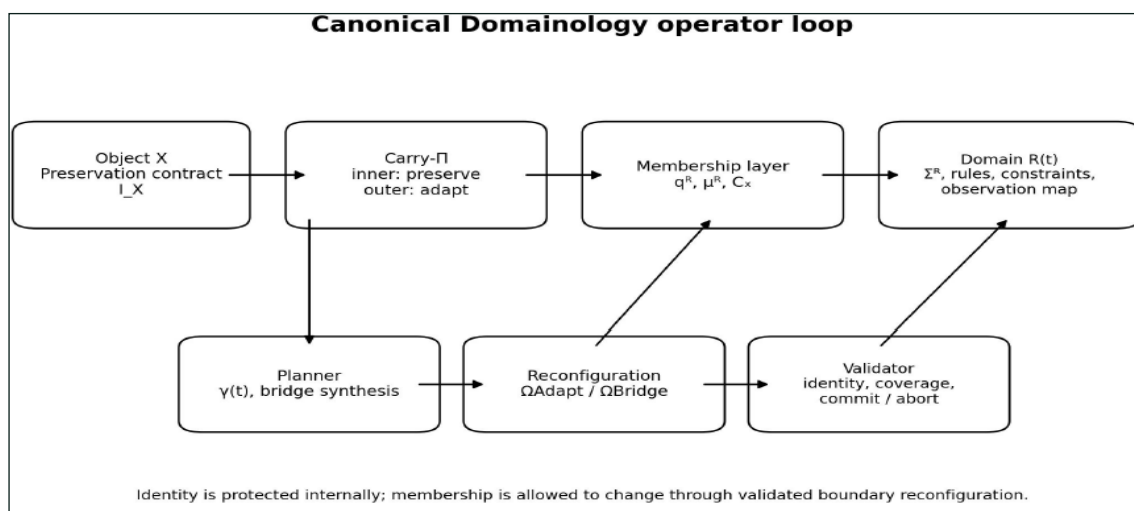


Figure 2: Canonical Domainology operator loop.

Canonical Operator Family

Domainology is operator-centered rather than vocabulary-centered. A domain concept becomes scientifically useful when it can be converted into an executable or testable operator with state variables, preconditions, invariants, actions, failure modes, and postconditions. This follows the broader operatorization principle [9].

The canonical family proposed here contains: Ω_{Seal} , which freezes the preservation contract and tolerances;

Ω_Open , which exposes or increases selected external degrees of freedom while keeping protected invariants sealed; Ω_Plan , which proposes a signature-space trajectory; Ω_Adapt , which changes the external boundary to track that trajectory; $\Omega_Handover$, which redistributes compatibility from source to target; Ω_Bridge , which synthesizes a temporary compatibility regime when the feasible relation is disconnected; $\Omega_Observe$, which maps the latent transition into observer-specific visibility; $\Omega_Validate$, which evaluates preservation, mismatch, coverage, and application constraints; and $\Omega_Commit/Abort$, which either finalizes target membership or halts/reverts without violating the inner contract.

The reference Domain Membership Operator is a composition of these stages rather than a single instantaneous reassignment: $\Omega_DM = \Omega_Commit \circ \Omega_Handover \circ \Omega_Adapt \circ \Omega_Plan \circ \Omega_Seal$, with validation running continuously. The important conceptual step is that membership can change gradually even if the target domain is discrete at the logical level. Raw compatibility and normalized membership provide a soft handover layer before commit.

A broader Domain Reconfiguration Operator Ω_DR acts not only on a state trajectory but on the feasible relation itself. If A_feas denotes the current set of admissible boundaries–domain configurations, $\Omega_DR:A_feas \rightarrow A_feas'$ may create, remove, or reshape feasible channels subject to explicit complexity and preservation costs. Path optimization then occurs in A_feas' , not merely in the original space. This operator distinguishes 'finding a route' from 'making a route admissible'.

A simple structural proposition follows. If source and target signatures lie in different connected components of a fixed feasible set and continuity is required, then no continuous path inside that unchanged feasible set connects them. The only logically available responses are to relax the constraints, introduce a valid intermediate regime, reconfigure the feasible relation, or declare the transition infeasible. Domainology treats those options explicitly rather than hiding them inside an unconstrained planner.

Table 2: Canonical Operator Family

Operator	Function
Ω_Seal	Freeze the preservation contract and tolerances.
Ω_Open	Expose selected external degrees of freedom without releasing protected invariants.
Ω_Plan	Construct a feasible signature trajectory.
Ω_Adapt	Track the planned trajectory by changing the external boundary.
$\Omega_Handover$	Redistribute compatibility from source to target.
Ω_Bridge	Synthesize an intermediate compatibility regime or aperture.
$\Omega_Observe$	Map latent transition to observer-specific outputs.
$\Omega_Validate$	Test preservation, mismatch, coverage, resources, and constraints.
$\Omega_Commit/Abort$	Finalize target membership or halt/revert safely.

$$\Omega_DM = \Omega_Commit \circ \Omega_Handover \circ \Omega_Adapt \circ \Omega_Plan \circ \Omega_Seal$$

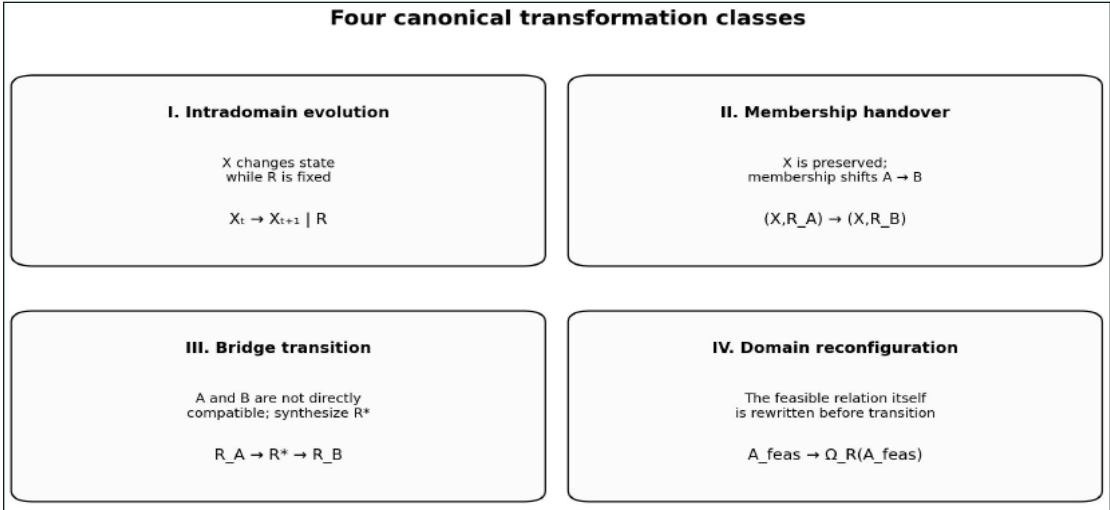


Figure 3: Four canonical transformation classes: intradomain evolution, membership handover, bridge transition and domain reconfiguration.

Research Branches of the Proposed Field

Structural Domainology studies how domains are represented, decomposed, compared, nested, and composed. Membership Theory studies raw compatibility, normalized membership, multi-membership, null membership, hysteresis, and commitment. Boundary Operatorics studies protected and adaptive boundary layers, openness variables, coupling control, and interface contracts. Transdomain Dynamics studies trajectories of changing membership. Domain Reconfiguration studies operators that alter the feasible relation or synthesize bridges. Domain Observability studies what different observers can infer from a latent transition. Prime-Structured Domainology studies discrete factorized representations for reconfiguration. Empirical Domainology develops measurable specializations and domain-specific tests. These branches are not intended as administrative categories. They are a research decomposition. A mature Domainology experiment should normally specify at least one object model, one domain signature, one membership rule, one preservation contract, one boundary model, one transition operator, and one validation protocol. Omitting any of these elements may be acceptable in exploratory work, but the omission should be explicit. This architecture also clarifies what Domainology is not. It is not a replacement for dynamical systems, control, topology, machine learning, or physics. It provides a higher-level language for problems in which the active domain relation is itself an object of transformation. In applications where the domain truly is fixed, ordinary methods remain sufficient and Domainology adds little.

Table 3: Proposed Research Branches

Branch	Primary Focus
Structural Domainology	Representation, decomposition, comparison, nesting, composition of domains.
Membership Theory	Raw compatibility, normalized membership, multi-membership, null membership, hysteresis.
Boundary Operatorics	Protected/adaptive interfaces, opening and sealing, coupling control.
Transdomain Dynamics	Trajectories of changing domain membership.
Domain Reconfiguration	Bridge synthesis and transformation of feasible relations.
Domain Observability	Observer-relative visibility, Dark Segments, inference limits.
Prime-Structured Domainology	Factorized integer representations for controlled rewrite.
Empirical Domainology	Measurable domain-specific specializations and experiments.

Prime-structured Domainology: Prime-TB Cut and Prime Excalibur

Prime-TB Cut and Prime Excalibur are included because audiences encountering a factorized Domainology implementation may otherwise mistake the prime representation for a metaphysical claim. In this paper they have a precise computational role. Let a discretized d-dimensional signature be $a=(a_1, \dots, a_d)$, with nonnegative integer coordinates. Assign distinct primes p_i to feature indices and encode $E(a)=\prod_i p_i^{a_i}$. By unique factorization, the exponent valuation $v_{\{p_i\}}(E)=a_i$ recovers every discrete coordinate exactly.

Prime-TB Cut is the factor-exponent extraction stage. It 'cuts' the composite integer representation into indexed prime-exponent coordinates. A rewrite operator then changes selected exponents according to the desired boundary update. Recomposition multiplies the updated prime powers to obtain the new encoded signature. Prime Excalibur is the full executable cycle: factorize $\rightarrow$ inspect $\rightarrow$ rewrite $\rightarrow$ recompose $\rightarrow$ validate.

For example, $a=(2,1,3)$ is encoded as $E=2^2 \cdot 3^1 \cdot 5^3=1500$. Prime-TB Cut returns $(v_2, v_3, v_5)=(2,1,3)$. If a boundary update requires the second coordinate to increase and the third to decrease, the rewrite may produce $a'=(2,4,1)$. Recomposition gives $E'=2^2 \cdot 3^4 \cdot 5=1620$. Nothing is lost on that integer grid because unique factorization makes the code injective.

The limitation is equally important. Continuous signatures must be quantized before prime encoding, so exactness applies only after discretization. The primes do not supply the semantic meaning of the features, and this representation alone does not show that a natural system is intrinsically prime-coded. PrimeDMO is therefore one implementation of $\Omega_{\text{Adapt}}/\Omega_{\text{DR}}$, not the definition of Domainology. Ordinary vectors, graphs, matrices, symbolic rules, or learned controllers may implement the same field-level operators.

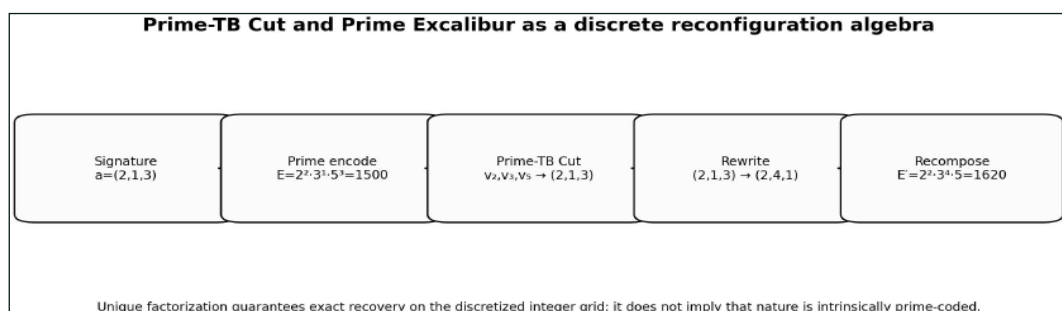


Figure 4: Prime-TB Cut and Prime Excalibur as a discrete reconfiguration algebra.

$$E(a) = \prod_i p_i^{a_i}, \quad v_{\{p_i\}}(E) = a_i$$

Domain Observability and Dark Segments

Domain membership can be observer-dependent even when the latent trajectory is continuous. Let $\gamma(t)$ be a latent transdomain trajectory and let $v_A(\gamma)$, $v_B(\gamma)$ be visibility scores for observers specialized to source and target domains. If both fall below their detection thresholds on an interval $[t_1, t_2]$, the record becomes $\text{visible}_A \rightarrow \text{unobserved} \rightarrow \text{visible}_B$ although $\gamma(t)$ itself has no discontinuity.

Domainology calls such an interval a Dark Segment. The name is deliberately observational rather than ontological. A Dark Segment does not establish disappearance from reality, tunneling, hidden dimensions, or any exotic mechanism. It establishes only that the chosen observation maps fail simultaneously over part of an otherwise modeled trajectory. Sensor dropout, occlusion, thresholding, and model error remain alternative explanations in empirical work.

This distinction is methodologically useful because apparent discontinuity is common in multi-regime systems. A rigorous analysis should therefore report the latent-state assumptions, observer maps, detection thresholds,

and the duration $\Delta_dark=t_2-t_1$. Predicting Δ_dark from a specified model is stronger than retrospectively labeling an unexplained gap.

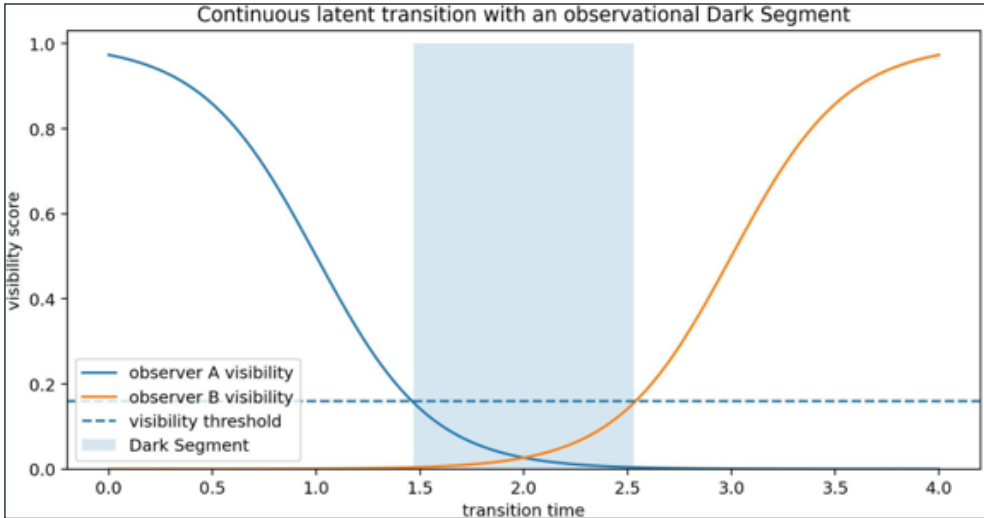


Figure 5: Observational Dark Segment: the latent trajectory is continuous while visibility windows do not overlap.

D-Bench V1.0: A Synthetic Benchmark for Domainology

To make the proposed field operational rather than purely terminological, this paper introduces D-Bench v1.0, a reproducible synthetic benchmark. Its purpose is internal discrimination: can the formal distinctions among direct reassignment, fixed-path handover, adaptive path planning, bridge synthesis, rate governance, and prime-structured representation be observed under controlled assumptions? It is not a physical experiment.

Five scenarios are used. D1 is a connected landscape with weak roughness. D2 contains a localized central incompatibility barrier. D3 contains a high wall spanning the permitted strip, so path planning alone cannot maintain required coverage. D4 commands an obstructed transition faster than the nominal boundary adapter can track. D5 adds a shifted major barrier and stochastic secondary obstacles. Five methods are compared: DirectJump, LinearHandover, AdaptivePath, BridgeDMO, and PrimeDMO.

Each method–scenario cell contains 40 deterministic pseudo-random replicates, producing 1,000 transition records. The master seed is 20260916. Continuous methods use the same first-order boundary adapter. AdaptivePath searches a smooth two-parameter path family. BridgeDMO may synthesize an aperture in D3 and invokes a rate governor in D4. PrimeDMO applies the same logic after quantizing the external signature to a 65-level grid and round-tripping it through prime-exponent encoding. The synthetic success rule is deliberately explicit: final target error <0.20, identity score ≥0.68, minimum coverage ≥0.10, and maximum boundary mismatch ≤0.60. The identity and cost functions are stress-test constructions rather than empirical laws. Their role is to make assumptions inspectable, reproducible, and easy to replace in future studies. All code, fixed settings, raw records, and generated figures are included in the supplementary package.

Table 4: D-Bench v1.0 scenarios.

Scenario	Construction	Primary test
D1	Connected; weak roughness	Basic handover
D2	Central Gaussian barrier	Safe planning vs straight path
D3	Wall spanning allowed strip	Bridge synthesis required
D4	Very short commanded duration	Adaptation-rate limit
D5	Shifted barrier + random obstacles	Robustness to landscape noise

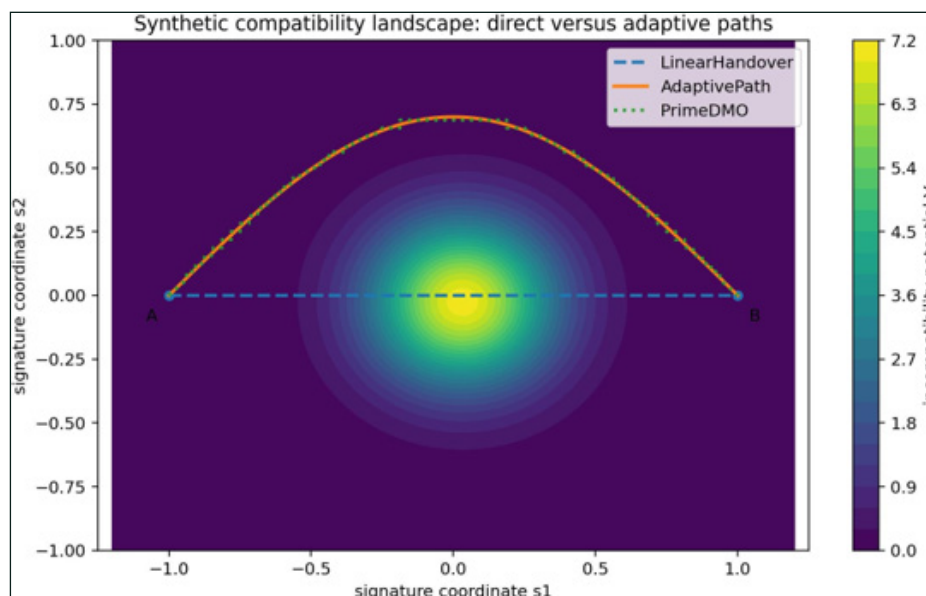


Figure 6: Central-barrier example: the straight path is short but highly exposed; adaptive and Prime DMO paths detour.

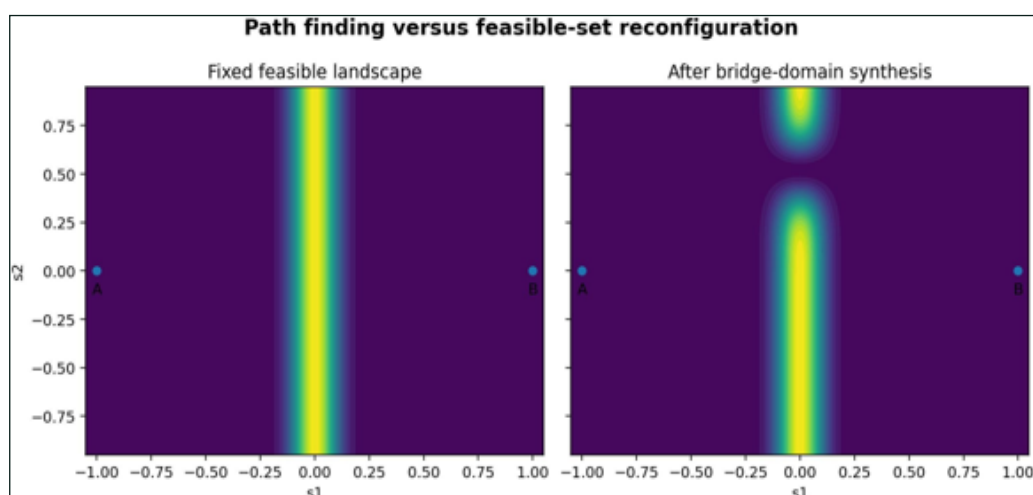


Figure 7: Path finding versus feasible-set reconfiguration: bridge synthesis creates a local compatibility aperture.

Benchmark Results

The results separate the operator classes cleanly. Direct Jump fails every scenario because the abrupt command generates a large boundary mismatch under the benchmark's continuity contract. Linear Handover succeeds in the connected case but fails the central barrier, the disconnected wall, and the fast transition; in the noisy landscape it succeeds in only 10% of replicates. Adaptive Path succeeds in D1, D2, and D5, demonstrating that trajectory optimization can suppress barrier exposure, but it fails D3 and D4 because geometry alone cannot create a missing compatibility channel or compensate for an adapter commanded beyond its rate limit.

BridgeDMO succeeds in all 200 of its benchmark runs. In D3 the decisive operation is not a better route through the original wall but bridge-domain synthesis, which creates a local compatibility aperture before planning. In D4 the rate governor extends the commanded transition duration, allowing the same first-order adapter to remain within the mismatch tolerance. Prime DMO also succeeds in all 200 runs. Its mean identity and coverage remain close to BridgeDMO, while path length and total cost are modestly higher because

quantization introduces stair-stepping.

Averaged over all five scenarios, BridgeDMO attains mean identity 0.739, maximum mismatch 0.420, minimum coverage 0.593, and total cost 3.132. Prime DMO yields 0.738, 0.446, 0.575, and 3.723 respectively. Adaptive Path has a 0.60 overall success rate; Linear Handover 0.22; Direct Jump 0.00. The numerical difference between BridgeDMO and Prime DMO is not evidence that primes are intrinsically superior. The defensible conclusion is representational: the prime code preserves enough discrete information to reproduce the reconfiguration logic under the chosen benchmark.

The benchmark therefore supports three field-level distinctions under synthetic assumptions. First, path adaptation and domain reconfiguration are not the same operation. Second, feasibility depends on boundary adaptation rate as well as geometric path quality. Third, a representation can change implementation cost without changing the higher-level operator contract. These distinctions are central to Domainology because they remain meaningful across computational, control, and potentially physical specializations.

Table 5: Success rate, 40 replicates per method–scenario cell.

Method	D1	D2	D3	D4	D5
Direct Jump	0.00	0.00	0.00	0.00	0.00
Linear Handover	1.00	0.00	0.00	0.00	0.10
Adaptive Path	1.00	1.00	0.00	0.00	1.00
Bridge DMO	1.00	1.00	1.00	1.00	1.00
Prime DMO	1.00	1.00	1.00	1.00	1.00

Table 6: Mean metrics across the five scenarios.

Method	Success	Identity	Max mismatch	Min coverage	Cost
DirectJump	0	0.014	2	0.1	5.646
LinearHandover	0.22	0.728	0.512	0.147	4.556
AdaptivePath	0.6	0.727	0.548	0.416	3.318
BridgeDMO	1	0.739	0.42	0.593	3.132
PrimeDMO	1	0.738	0.446	0.575	3.723

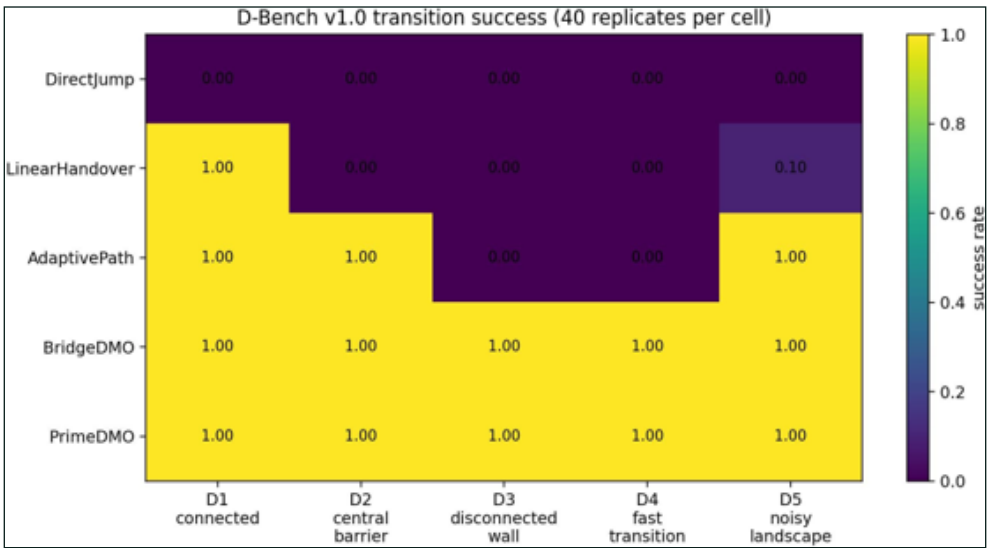


Figure 8: D-Bench v1.0 success-rate heatmap.

Relation to Adjacent Disciplines

Viability theory and invariant-set methods ask whether a state can remain inside admissible constraints [1,2]. Domainology inherits this concern but allows the compatibility target and, through reconfiguration, parts of the feasible relation to change. Hybrid systems provide natural tools for seal, commit, abort, and mode-dependent dynamics [3]. Domainology differs in making 'which domain accepts the object' an explicit relation rather than only a mode index.

Path planning searches for trajectories in a given configuration space [5]. Domainology uses planning but separates it from bridge synthesis and feasible-set rewriting. Graph and model transformation provide algebraic views of changing structural relations; a domain-membership handover can be represented as controlled rewiring between an object node and domain nodes. Domain adaptation studies distributional mismatch in machine learning; Domainology generalizes the vocabulary of source–target compatibility but is not limited to statistical learning [6-8].

Ontology and knowledge representation ask what entities and relations a model contains. Domainology asks a more operational question: under what conditions can a relation of membership, admissibility, coupling, or observation be changed while a selected identity contract remains valid? The distinction is analogous to the difference between describing a graph and specifying a validated graph rewrite.

The proposed field should therefore be judged by whether it produces useful operators, theorems, benchmark tasks, measurement protocols, and empirical specializations—not by whether the word 'domain' appears in many existing disciplines. Its contribution lies in centering a particular transformation problem and building a reusable formal language around it.

Table 7: Positioning relative to adjacent disciplines.

Adjacent field	Typical focus	Domainology adds
Viability / invariance	Fixed or controlled admissible sets	Membership relation and boundary can be explicit state variables.
Hybrid systems	Continuous flow + discrete switching	Adds validated domain-membership handover and reconfiguration.
Path planning	Find a path in a given space	Separates path finding from changing the feasible space.
Graph transformation	Rewrite structural relations	Specializes rewrites around persistent object–domain relations.
Domain adaptation	Source–target statistical mismatch	Generalizes beyond ML and treats compatibility as dynamic.
Ontology / KR	Describe entities and relations	Focuses on executable, validated transformation of membership relations.

Falsifiability, Limitations and Claim Boundary

A Domainology model is falsifiable at the level at which it is instantiated. A proposed transition fails if no admissible trajectory and boundary configuration satisfy the preservation, mismatch, coverage, resource, and domain-specific constraints. A bridge operator fails if its complexity exceeds an allowed budget or if its synthesized regime cannot be independently realized. An observational account fails if predicted visibility windows do not match measured ones. A prime implementation fails if its discretization error or computational cost destroys the claimed behavior.

The present benchmark has important limitations. It is two-dimensional, low-order, and synthetic. The path

family is deliberately simple. The bridge operator modifies a known potential landscape instead of learning an interface from first principles. The success thresholds and identity function are benchmark design choices. The prime implementation uses only a 65-level discretization. These limitations are features of a first benchmark, not evidence that the same performance will hold in high-dimensional or physical systems.

Most importantly, Domainology is not evidence for teleportation, travel between realities, switching off gravity, disappearing from physical reality, or any extraordinary transport mechanism. Such claims require measurable physical variables, conservation-law accounting, instrumentation, preregistered tests, comparison with null models, and independent replication. The framework can help state such hypotheses more precisely, but a language for a hypothesis is not evidence for the hypothesis.

The strongest current claim is therefore methodological: treating domain membership and feasible-domain relations as explicit stateful objects yields a coherent operator language that distinguishes preservation, handover, planning, reconfiguration, and observation. The D-Bench results show that these distinctions are computationally nontrivial under the supplied assumptions.

Table 8: Claim boundary.

Status	Statement
Supported	Domains, membership, boundaries, and reconfiguration can be formalized as testable objects/operators.
Supported	BridgeDMO and PrimeDMO outperform specific baselines in the supplied synthetic benchmark.
Supported	Prime-factor encoding is injective on the discretized integer signature grid.
Not supported	Prime Excalibur switches off gravity, teleports matter, or removes matter from reality.
Not supported	An observational Dark Segment by itself proves quantum tunneling or a hidden physical domain.

Research program and open problems

A field becomes useful only if it generates difficult questions. The first program is high-dimensional Domainology. Real domain signatures combine continuous variables, categorical modes, graphs, sets, and delayed observations. Product-space metrics, reachability analysis, viability kernels, and model-predictive reconfiguration should replace the present two-dimensional toy geometry.

The second program is bridge complexity. A bridge should never be free. Future work should define a functional K_{bridge} that penalizes the number of rewritten signature components, magnitude of change, persistence time, energy or compute cost, information loss, and reversibility. This prevents the trivial solution in which every impossible transition is declared possible by granting an arbitrarily powerful bridge operator.

The third program is operator composition. If $\Omega_{\text{DM}}(A \rightarrow B)$ and $\Omega_{\text{DM}}(B \rightarrow C)$ are feasible, when is their composition equivalent to a direct $A \rightarrow C$ operator? The answer depends on path dependence, hysteresis, noncommuting boundary rewrites, and residual bridge state. This suggests a compositional algebra of domain operators and, eventually, a category-like treatment in which objects are domain-coupled systems and morphisms are validated membership transformations.

The fourth program is empirical Domainology. The recommended order is conservative: software/container migration, communication adapters, simulated robots crossing controller regimes, hardware-in-the-loop switching, and measurable physical interfaces should be studied before extraordinary claims. Each specialization should define observable signatures, preservation variables, costs, and failure criteria. A successful field must

survive ordinary tests before speculative applications carry weight.

The fifth program is automated operator discovery. Prime Excalibur and other structured representations can be used as search spaces for finding sparse reconfiguration rules. Scientific machine learning may then search not only for a state equation but for the smallest executable operator that transforms one admissible domain relation into another while preserving declared invariants. This links Domainology to operator-centric machine intelligence without making any one machine-learning architecture foundational.

Conclusion

Domainology is proposed as a general research discipline for problems in which domains, membership, boundaries, and admissible relations are themselves part of the dynamics. Its core move is simple: stop treating the domain as silent background when the scientific question concerns changing the relation between an object and that domain [10,11].

The field inherits rigorous tools from viability, hybrid systems, observability, planning, graph transformation, and domain adaptation, but reorganizes them around a first-class membership relation. The Domain Membership Operator provides a reference transition protocol; Carry-II separates protected internal structure from adaptable external coupling; bridge synthesis distinguishes path finding from feasible-set creation; Domain Observability accounts for Dark Segments; and Prime-TB Cut/Prime Excalibur provide one exact discrete algebra for decomposition, rewrite, and recomposition after quantization.

D-Bench v1.0 demonstrates that these distinctions can be encoded and tested. Within the synthetic benchmark, adaptive paths solve obstacle problems that defeat straight handover, bridge synthesis solves a disconnected-feasibility problem that path optimization cannot, rate governance recovers transitions that exceed adapter speed, and prime encoding reproduces the reconfiguration behavior with modest quantization overhead.

The proposal is therefore not a claim that a new physical force has been discovered. It is a proposal for a new object of systematic study: the dynamics of belonging to, leaving, bridging, observing, and reconfiguring domains while selected identity contracts persist. In its shortest form, Domainology asks: when the state changes, what remains the same—and when the domain changes, what makes the transition valid?

Declarations

Conflict of interest. The author reports no known competing interests. Funding. No external funding is reported.

Data and code availability. All benchmark code, synthetic records, fixed settings, and figures are included in the accompanying supplementary package. Research integrity. D-Bench is synthetic by design and validates behavior only under the encoded assumptions.

Generative AI. Generative AI tools were used to assist language drafting, code prototyping, numerical checking, and figure layout; the author is responsible for the conceptual framework, interpretation, verification decisions, and final manuscript.

References

1. Aubin JP (1991) Viability Theory. Birkhäuser.
2. Blanchini F (1999) Set invariance in control. Automatica 35: 1747-1767.
3. Goebel R, Sanfelice RG, Teel AR (2012) Hybrid Dynamical Systems: Modeling, Stability, and Robustness. Princeton University Press.
4. Hermann R, Krener AJ (1977) Nonlinear controllability and observability. IEEE Transactions on Automatic

Control 22: 728-740.

5. LaValle SM (2006) Planning Algorithms. Cambridge University Press.
6. Ehrig H, Ermel C, Golas U, Hermann F (2015) Graph and Model Transformation: General Framework and Applications. Springer.
7. Ben David S, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, et al. (2010) A theory of learning from different domains. Machine Learning 79: 151-175.
8. Ben David S, Lu T, Luu T, Pal D (2010) Impossibility Theorems for Domain Adaptation. Proceedings of Machine Learning Research 9: 129-136.
9. Trung PT (2026) Operatorization: A Framework for Transforming Fixed Solutions into Executable Knowledge. SSRN.
10. Trung PT (2026) Domain Membership Operators: Identity-Preserving Transitions Across Heterogeneous Dynamical Domains. Working manuscript.
11. Trung PT (2026) Prime Excalibur: A Falsifiable Operator-Centric Scientific Machine Learning Framework for Sparse Structure Discovery. SSRN working paper.

Copyright: ©2026 Phan Thành Trung. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.