Directional–Deformation–Dissipation Architecture (D^3A)

A Structural Layer for Adaptive Systems by Maksim Barziankou

1 Executive Statement

This document introduces and fixes a structural architectural layer of adaptive systems that is not reducible to optimization, nor to control, nor to learning, nor to objective-based or reward-based formulations. The discussion concerns a layer that is present in all adaptive systems—physical, computational, biological, and cyber-physical—but typically remains unseparated and unformalized.

The key statement of this architecture is the following: directed change always induces structural deformation before it becomes observable as translation or correction, and the long-term stability of a system is determined not by how well it minimizes error, but by how it regulates, accumulates, and bounds this deformation over time.

Within D^3A, adaptation is treated not as a local act of correction or optimization, but as a process that has geometry, asymmetry, dissipation, and irreversible cost. These characteristics are not side effects of algorithms; they form an independent layer that must be explicitly represented, measurable, and governable.

D^3A does not propose a new controller, a learning algorithm, or an objective function. The architecture is deliberately positioned orthogonally to existing methods. It defines a set of primitives and their causal order, within which such methods may be applied, but which they themselves do not cover and do not describe.

The purpose of this document is to fix this layer in canonical form: to define its basic primitives, their admissible interrelations, the boundaries of interpretation and correct use, and to provide a stable architectural framework for engineering implementation, analysis, and standardization.

2 Scope and Non-Scope

2.1 Scope

The scope of this specification covers any systems that change over time, interact with an environment, and apply directed influences—whether physical impulses, computational updates, an agent's decisions, or distributed actions in a network. What is essential is not a particular substrate, but the fact of directed change under conditions of uncertainty, delay, or complex environmental dynamics.

Within this scope, D^3A formalizes several fundamental aspects of adaptive behavior. First, the concept of the direction of admissible change is introduced, independent of its magnitude. Second, the fact is fixed that any directed influence induces structural deformation that is not reducible to scalar error or energy. Next, the role of adaptive dissipation is defined, suppressing destructive components of deformation without loss of responsiveness. Substantial attention is given to the accumulation of residual deformation, or drift, as an inevitable cost of prolonged adaptation. Finally, mechanisms of structural bounding of expansion, explicit alignment of the system with the environment, and regulation of internal time on the basis of the causal reliability of interaction are introduced.

This architecture is applicable in robotics, autonomous agents, control systems, distributed computing, and human–machine interfaces; however, it is not tied to any one of these domains and does not use their domain-specific assumptions.

2.2 Non-Scope

This specification in principle does not define and does not replace existing algorithmic mechanisms. It is not a controller, a control law, a loss function, a method of optimization, a learning scheme, or a reward model. It also does not introduce psychological, cognitive, or metaphysical notions as causal elements of the architecture.

D^3A does not compete with classical and modern methods of control and learning and does not claim their generalization. On the contrary, it describes a layer that these methods cross implicitly, but usually do not make explicit. Attempts to reduce D^3A primitives to error, noise, reward, or algorithmic parameters are considered as going beyond the bounds of correct interpretation of this architecture.

2.3 Separation of Layers

Critically important is the strict separation between the structural (physical) layer described in the main text and the interpretive or philosophical interface intended exclusively for human understanding. The main text of the specification is fully self-sufficient and does not rely on concepts of intention, will, agency, or meaning.

Any such concepts, if they are used, are placed into a separate appendix and have no formal, causal, or normative status in the architecture. Mixing these levels leads to a loss of engineering precision and undermines the very purpose of the specification.

3 Canonical Primitives

This section fixes the canonical set of primitives of the D^{3}A architecture. These primitives are minimal, irreducible, and mutually necessary for describing the structural layer of adaptive systems. None of them presupposes a particular implementation, algorithm, or substrate; each is defined through its role in the causal structure of adaptive change.

3.1 Directional Constraint

Any adaptive system changes not arbitrarily, but within the bounds of some set of admissible directions of change. Direction in this context is not a control vector in the usual sense and does not set the magnitude of influence. It determines the form and orientation of possible state change relative to the current structure of the system and the environment.

Directional Constraint is an operator that, from the current state of the system and the state of the environment, determines the set of admissible directions of further change. This operator may be realized through invariants, safety constraints, geometric or topological conditions, semantic or structural constraints; however, its essence does not depend on the particular form of realization.

The key point is that Directional Constraint precedes any act of control or learning. It does not correct the result and does not optimize deviation, but determines which directions of change are considered admissible at all. Any adaptation that occurs outside this set is considered structurally incorrect regardless of its short-term effectiveness.

3.2 Directed Perturbation and Jerk

When a system applies an admissible direction of change, this change is realized as a directed perturbation—an action, an update, an impulse, or a transition. In real adaptive systems such changes are never ideally smooth. Their interaction with constraints, the environment, and internal structure produces high-frequency components of change that carry structural information.

To reveal this information, a representation of a jerk-like component is introduced—a difference or derivative of directed changes in time. This quantity is not interpreted as physical jerk in the narrow sense, but serves as an indicator of sharpness and non-uniformity of directed influence in the chosen representation.

Directed Perturbation and the associated jerk component form the input to the next primitive of the architecture, allowing one to distinguish simple translational change from structurally loaded change.

3.3 Operational Spin

Operational Spin fixes a fundamental fact: directed change induces antisymmetric structural deformation in the representation of the system or the environment before it manifests as observable displacement or correction.

Spin is defined as the antisymmetric part of the local gradient of directed change or of its jerk component. It is not an error, noise, or energy and is not reducible to them. Spin is a geometric characteristic reflecting a violation of local isotropy caused by directed influence.

It is important to emphasize that Operational Spin is not an artifact of a particular substrate. It is equally definable in continuous physical media, discrete computational spaces, graphs, parametric spaces, and distributed systems. Wherever directed change and local difference exist, an antisymmetric component of deformation arises.

Spin is the first structural trace of directed influence and serves as an input quantity for mechanisms of adaptive regulation.

3.4 Adaptive Dissipative Response (ΔE-class)

Operational deformation fixed as Spin is not in itself pathological. Adaptive systems inevitably induce structural load. What is critical is not its presence, but the manner of its regulation.

Adaptive Dissipative Response is a mechanism that suppresses destructive components of deformation while preserving the system's ability to respond to changes in the environment. In the D^3A architecture this mechanism is designated as ΔE-class, emphasizing its thermodynamic rather than optimization nature.

ΔE-class regulation does not minimize Spin and does not seek to drive deformation to zero. It adaptively reduces its intensity depending on the observed structural load, variability, and instability, bringing the system into a regime of governed dissipation. In this way responsiveness is preserved without accumulation of uncontrolled stress.

3.5 Residual Spin and Drift

Even in the presence of adaptive dissipation, part of the structural deformation remains inevitable. This residual deformation, or residual spin, does not disappear and is not compensated automatically. It accumulates over time as structural drift.

Drift in the D^3A architecture is defined as an accumulated measure of residual antisymmetric deformation associated with changes in internal operators, representations, or rules of adaptation of the system. It reflects the irreversible cost of prolonged existence and adaptation.

It is fundamentally important that Drift is not noise, error, or a random effect. It represents the structural memory of the system about its own changes. Any system with nonzero adaptation and finite dissipation has nonzero Drift.

3.6 Structural Bounds on Expansion

Since Drift is irreversible, the D^3A architecture introduces the primitive of structural bounding of expansion. An adaptive system cannot expand its space of states, parameters, or representations indefinitely without destroying its own identity.

Structural Bounds define an internal threshold of admissible accumulated drift. Upon reaching this threshold, the system must change its mode of existence: stop further expansion, initiate restructuring, or transition to another level of representation. These bounds are formed not by external rules, but by internal structural invariants of the system.

Thus, D^3A introduces a fundamentally new type of constraint—ontological, not operational.

3.7 Alignment Variables

In many adaptive systems, not only the internal state is critical, but also the degree of concordance of the system with the environment. For this purpose explicit alignment variables are introduced—alignment variables.

Alignment in the D^{3}A architecture is not reducible to error, reward, or energetic efficiency. It is defined through measurable characteristics of interaction between the system and the environment and is used as an independent object of control. Through alignment the system may regulate regimes of contact, adhesion, stability, or cooperation without increasing effort or suppressing response.

This primitive forms a bridge between the structural state of the system and the dynamics of its interaction with the external world.

3.8 Adaptive Internal Time

The last canonical primitive is the internal time of the system. In the D^{3}A architecture time is not treated as a fixed external scale. Instead, the system possesses its own internal time, the rate of which adapts depending on the causal reliability of interaction with the environment.

When reliability decreases—with increases in delays, phase mismatches, or uncertainty—internal time slows down, which prevents regime chatter and structural overload. Under high reliability internal time accelerates, allowing the system to adapt rapidly.

Adaptive Internal Time completes the set of primitives of D^{3}A, ensuring coordination of all previous layers in time.

4 Layered Architecture and Causal Ordering

The D^{3}A architecture is determined not only by the set of canonical primitives, but also by their strict causal ordering. These primitives are not equal modules that may be arbitrarily rearranged, included, or excluded. Each arises as a consequence of the previous and loses meaning outside this order.

Violation of the causal order leads not merely to an incorrect implementation, but to a change of the descriptive layer, in which the behavior of the system no longer belongs to the D^{3}A architecture, even if similar terms or formulas are used.

The architecture is organized as a sequence of levels, each of which introduces a new type of structural fact that is impossible at the previous level.

L0. Direction

The base level of the architecture is the direction of admissible change. At this level it is determined where the system is permitted to change, regardless of how quickly or with what intensity this occurs. Direction sets the form of possible evolution and thereby precedes any acts of control, learning, or correction.

Without an explicit Direction level it is impossible to distinguish correct adaptation from structural destruction, since any change will be evaluated exclusively by its local effect. All subsequent levels of the architecture assume the existence of Direction as an initial condition.

L1. Deformation (Operational Spin)

As soon as direction is realized as a directed influence, structural deformation arises. This level fixes the fact that directed change is not a pure translation, but is accompanied by antisymmetric deformation of the representation of the system or the environment.

Operational Spin arises inevitably and does not depend on the choice of algorithm or substrate. It is the first measurable structural trace of the realization of direction and cannot be correctly defined without the preceding Direction level. Attempts to introduce Spin without explicit direction lead to a substitution of deformation by noise or randomness.

L2. Dissipation

The Dissipation level introduces regulation of deformation. Here the system does not attempt to eliminate deformation, but governs its intensity and distribution while preserving the ability to respond. Dissipation has no meaning without explicitly defined Spin, since without it there is no object of regulation.

At this level the distinction arises between destructive and admissible deformation. Any regulation that does not rely on measurable structural deformation belongs to another class of methods and does not realize the D^3A architecture.

L3. Drift and Structural Memory

Even in the presence of adaptive dissipation, part of the deformation remains and accumulates. This level fixes the emergence of the system's structural memory about its own changes. Drift is not a state in the usual sense; it reflects a history of irreversible deformations embedded in the very structure of adaptation.

Drift is impossible without preceding levels, since it is defined as an integral of residual deformation. Attempts to treat Drift as noise, error, or a parameter of learning ignore its causal origin and take the system beyond the bounds of the D^3A architecture.

L4. Alignment

At the Alignment level the architecture goes beyond the internal dynamics of the system and introduces explicit regulation of concordance of the system with the environment. Alignment relies on the already existing structural memory and current deformation and therefore cannot be correctly realized as a local function of error or reward.

Alignment defines the mode of interaction of the system and the environment, including adhesion, stability, and cooperation. Its correct operation is impossible without an understanding of accumulated Drift, since it is Drift that determines admissible and safe modes of concordance.

L5. Internal Time

The final level of the architecture is internal time. At this level the system ceases to perceive time as an external, uniform scale and begins to regulate its own temporal dynamics depending on the causal reliability of interaction.

Internal Time closes the architecture, since it binds Direction, Deformation, Dissipation, Drift, and Alignment into a single dynamics. Without internal time the previous levels cannot be coordinated in time, and the system inevitably encounters oscillations, regime chatter, and structural overload.

Architectural Closure

The causal order L0–L5 is rigid. No level may be omitted, rearranged, or realized after the fact without destruction of architectural integrity. The use of individual primitives outside this order is admissible, but in that case the system does not realize the D^3A architecture and should not be described in its terms.

This section fixes the architectural closure of the D^3A layer and defines the minimal conditions under which an implementation may be considered valid.

5 Valid and Invalid Compositions

The D^{3}A architecture admits many implementations, but does not admit arbitrary interpretations. Correctness of an implementation is determined not by whether the system demonstrates operability or improved metrics, but by whether the causal order and the semantics of the canonical primitives are observed.

This section fixes the conditions under which the composition of primitives is considered architecturally correct, as well as cases in which the use of similar notions leads beyond the bounds of the D^{3}A layer, even if externally the behavior of the system appears consistent.

5.1 Valid Composition Principle

A composition is considered valid if and only if each used primitive is introduced after all primitives that causally precede it, does not redefine the semantics of an earlier level, and is not reducible to a derived quantity of another level.

In other words, a primitive may be realized in different ways, but it cannot be replaced by a function of another primitive without loss of architectural correctness.

Directional Constraint cannot be realized as a function of error or reward. Operational Spin cannot be extracted from variance or noise. Drift cannot be replaced by a moving average of parameters. Alignment cannot be reduced to local optimality. Internal Time cannot be equivalent to changing an integration step without accounting for causal reliability.

If these conditions are violated, the system ceases to realize the D^{3}A layer, even if individual terms are used correctly.

5.2 Invalid Reorderings

Any composition is considered incorrect in which regulation (Dissipation) is introduced without an explicit object of deformation (Spin), deformation is evaluated without a fixed admissible direction (Direction), accumulation of Drift is interpreted without indicating residual deformation, Alignment is used without accounting for accumulated Drift, or Internal Time is applied without explicit signs of causal unreliability.

Such reorderings destroy the causal structure of the layer and move the description of the system into another class, even if formulas or blocks look similar.

5.3 Reduction Errors

A class of reduction errors is singled out, in which D^{3}A primitives are reduced to more habitual quantities. These errors are architecturally incorrect regardless of practical effect.

Spin is not a form of regularization and cannot be correctly realized as a penalty in a loss function. Drift is not noise, forgetting, or overfitting. Alignment is not reward or an objective function. Internal Time is not heuristic slowing or speeding of updates without an explicit criterion of causal reliability.

Such reductions erase the distinction between structural and parametric levels and deprive the architecture of its subject.

5.4 Partial Adoption

The use of individual D^{3}A primitives in isolation is admissible, but must be treated as partial borrowing, not as realization of the architecture as a whole. In such cases a system may be described as using elements of Direction, Spin, or Drift, but not as realizing the D^{3}A layer.

This boundary is principled. The D^{3}A architecture is not a brand and not a method, but a coherent causal construction. Partial coincidence of terms or formulas is not a sufficient basis to claim architectural equivalence.

5.5 Architectural Boundary

This section introduces a clear boundary between correct use of the D^{3}A architecture and other approaches. This boundary does not evaluate the quality or effectiveness of alternative methods, but fixes whether they belong to this structural layer or not.

In this way the main form of blurring of architectural ideas is eliminated: the use of familiar words with altered semantics.

6 Compatibility and Non-Subsumption

The D^{3}A architecture is designed so as to be compatible with a broad class of existing methods of control, learning, and optimization, without entering into competition with them and without substituting their internal mechanisms. At the same time, this compatibility does not mean inclusion or reducibility of D^{3}A to these methods.

The key distinction lies in the level of description. Classical and modern methods, including regulators, filters, optimizers, and learning algorithms, operate the parametric or behavioral level of the system. D^3A, by contrast, describes the structural level, which precedes the choice of a specific mechanism of action and persists under its replacement.

For this reason D^3A may be used jointly with various methods, but cannot be fully described or replaced by them.

6.1 Classical Control Systems

Classical regulators such as PID, LQR, or MPC may operate within the D^3A architecture without any change to their basic structure. In this case Direction sets the admissible domain of application of the control influence, Dissipation regulates structural load, and Internal Time coordinates the tempo of the regulator with the reliability of feedback.

However, classical regulators by themselves do not define Direction, do not measure Spin, do not accumulate Drift, and do not introduce structural bounds on expansion. Consequently, they do not cover the D^3A layer and cannot be considered its special case.

6.2 State Estimation and Filtering

Methods of state estimation and filtering, including Kalman and Bayesian approaches, are compatible with the D^3A architecture in the part of state representation and uncertainty. They may be used to build observers to which quantities defined at the structural level are provided as inputs.

Nevertheless, such methods operate statistical uncertainty and posterior estimates, not structural deformation. They do not fix the antisymmetric component of directed change and do not distinguish reversible and irreversible adaptation. Therefore, they do not substitute for and do not realize the D^3A primitives.

6.3 Learning and Reinforcement Learning

Learning algorithms, including reinforcement learning, may use the D^3A architecture as a structural frame. Direction may constrain the space of admissible actions, Dissipation may suppress destructive learning regimes, Drift may fix the cost of long-horizon adaptation, and Structural Bounds may prevent degradation of representations.

At the same time D^3A does not introduce a reward function, does not define a learning strategy, and does not optimize expected utility. Any attempt to interpret Alignment as reward or Drift as learning regularization is a reduction and goes beyond the bounds of the architecture.

6.4 Optimization and Objective-Based Methods

Optimization methods, including gradient and heuristic approaches, may be applied within constraints set by D^{3}A. In this case the architecture determines which directions of optimization are admissible, what the structural cost of change is, and when further optimization must be stopped or reformulated.

However, optimization by definition is oriented to an objective function and does not distinguish the structural consequences of achieving a goal. Therefore it cannot replace a layer in which these consequences are the subject of explicit accounting.

6.5 Non-Subsumption Principle

From the stated compatibility follows the principle of non-subsumption. None of the existing classes of methods includes the D^{3}A architecture as a special case, since none of them fixes simultaneously direction, deformation, dissipation, accumulated drift, structural bounds, alignment, and internal time as an interrelated causal structure.

Similarly, the D^{3}A architecture does not claim replacement of these methods. It defines a level at which their correct, stable, and long-lived application becomes possible.

7 Misuse and Invalid Interpretations

The D^{3}A architecture is vulnerable not to incorrect implementation, but to semantic substitution. In practice this manifests when correct terms are used to describe mechanisms different in nature. Such an approach may produce locally acceptable results, but destroys the subject of the architecture itself and makes further reasoning incorrect.

First of all, it is inadmissible to treat Directional Constraint as a function of error, reward, or preferences. The direction of admissible change is not derived from an objective function and is not a consequence of optimization; it sets the form of admissible evolution prior to any computations of effectiveness. Substitution of Direction by the result of optimization removes the distinction between admissibility and expediency and takes a system beyond the bounds of D^{3}A.

Operational Spin cannot be interpreted as noise, variability, variance, or a regularizer. Spin fixes antisymmetric deformation arising from directed influence and therefore is not reducible to symmetric statistical measures. The use of penalties in a loss function as an "equivalent" of Spin is a reduction and is not considered a correct realization of the architecture.

Adaptive Dissipative Response (ΔE-class) is not a minimization of deformation and is not identical to smoothing. Dissipation governs the distribution and intensity of structural load without eliminating it. Any realization in which the goal is to drive deformation to zero or to suppress response contradicts the meaning of this primitive.

Drift does not permit interpretation as noise, forgetting, overfitting, or a side effect of learning. It represents accumulated irreversible deformation and is the structural memory of a system. Reducing Drift to a statistical effect deprives the architecture of a long-horizon dimension and makes it impossible to introduce structural bounds correctly.

Alignment cannot be reduced to reward, error, or efficiency. Alignment describes the mode of concordance of a system with the environment and is used to regulate interaction, not to optimize a goal. Substituting Alignment by a reward function erases the boundary between structural and behavioral levels.

Internal Time is not equivalent to heuristic changing of an integration step or an update frequency. Internal time is regulated by causal reliability of interaction and must be linked to delays, phase mismatches, and uncertainty. Any realization that does not rely on these factors does not belong to D^{3}A.

Finally, it is inadmissible to use individual D^{3}A primitives as metaphors or heuristics without explicit indication that the architecture as a whole is not realized. Partial borrowing is possible, but requires explicit delimitation to avoid false equivalence.

8 Reference Use and Architectural Neutrality

The D^{3}A architecture is intentionally formulated as neutral with respect to domains of application. This means that its primitives and causal order do not depend on physical units, the dimensionality of space, the discreteness of time, or a particular substrate.

In contact interaction systems D^{3}A allows one to separate admissible directions of motion, structural deformation of contact, and accumulated cost of interaction, which provides stability without excessive effort. In autonomous agents the architecture sets structural bounds on learning and prevents degradation of representations under long-horizon adaptation. In distributed and networked systems D^{3}A allows one to distinguish delays as an informational signal and as a source of structural load, coordinating the tempo of nodes without global synchronization. In human–machine interfaces the architecture provides concordance of system response with the user's dynamics without suppressing sensitivity.

These examples do not exhaust the scope of application and do not claim optimality. They serve a single purpose: to show that D^{3}A fixes a layer common to different classes of systems and is not tied to a particular implementation or industry.

It is important to emphasize that the architecture does not prescribe how a system must be implemented and does not evaluate the effectiveness of specific solutions. It determines which structural facts must be represented and coordinated for adaptation to remain stable and long-lived.

A Interpretive Interface (Non-Normative)

This appendix intentionally omits any operational or implementational detail.

This appendix does not introduce new primitives, does not expand formal definitions, and does not change the causal order of the D^{3}A architecture. Its sole purpose is to provide an interpretive interface for human understanding of those structural facts that are described in the main text in engineering terms.

Any notions used in this appendix have no normative status and must not be applied for formal analysis, implementation, or verification of the architecture.

A.1 Directed Change as Structural Commitment

In an interpretive perspective, Directional Constraint may be understood as a form of the system's structural commitment with respect to its own future. The choice of direction means not merely movement toward a result, but acceptance of which changes the system considers admissible for itself. This commitment precedes evaluation of effectiveness and cannot be derived from it.

It is important to emphasize that this interpretation does not ascribe intentions or goals to the system. It serves only to separate admissibility of change from its usefulness in intuitive terms.

A.2 Deformation as the Cost of Acting

Operational Spin and the associated deformation may be understood as the cost of the fact of acting itself, regardless of whether the desired effect was achieved. In this perspective any directed influence leaves a trace in the structure of the system or the environment, even if it was formally correct and locally successful.

This interpretation helps avoid a common error in which an action is evaluated only by the result, and not by the structural consequences of its execution.

A.3 Dissipation as Preservation, Not Suppression

Adaptive Dissipative Response in an interpretive sense may be regarded as a mechanism of preserving the viability of a system. Dissipation does not suppress activity and does not eliminate change; it prevents the transition of changes into destructive regimes.

Here the distinction between "to do less" and "to do otherwise" is important. D^3A formalizes precisely the second approach.

A.4 Drift as Memory of Becoming

Drift, in an interpretive frame, may be understood as the system's memory of what it has become in the process of adaptation. This is not a memory of events and not an archive of data, but a trace of irreversible structural changes accumulated over time.

Such a perspective emphasizes that long-horizon adaptation always has a cost, even if it does not manifest in short-horizon metrics.

A.5 Structural Bounds as Self-Limitation

Structural Bounds may be intuitively understood as a form of self-limitation of a system. Unlike external constraints, they arise from internal structure and reflect the limits beyond which further change ceases to be preservation of the system and becomes its destruction.

This interpretation does not introduce teleology or agency, but only emphasizes the distinction between growth and loss of identity.

A.6 Alignment as Mode of Coexistence

Alignment in an interpretive sense describes a mode of coexistence of a system and an environment. The discussion is not about achieving a goal and not about maximizing benefit, but about a form of interaction that remains stable over time.

This allows one to separate alignment from optimization in intuitive terms and to avoid reducing Alignment to a reward function.

A.7 Internal Time as Responsiveness

Adaptive Internal Time may be understood as a measure of a system's responsibility for its own tempo of change. Slowing and speeding of internal time reflect not computational preferences, but an assessment of the reliability of the connection with the world.

Such a perspective helps to see that time in D^{3}A is not a resource, but a regulator of stability.

A.8 Final Note on Interpretation

All given interpretations are intended exclusively to facilitate human understanding of the architecture. They do not add new requirements, do not replace formal definitions, and cannot be used to prove correctness of an implementation.

The D^{3}A architecture remains fully defined by the main text of the document.

10.5281/zenodo.18041361

MxBv
Poznan, 2025