

Is System Dynamics Made of DSRP?

Testing Whether the Organizational Structure of System Dynamics Is Contained Within DSRP-484

Derek Cabrera ^{a,b,@}

a: Cabrera Research Lab

@: dac66@cornell.edu

ABSTRACT: System dynamics (SD) is among the most formally developed traditions in systems thinking and is sometimes conflated with systems thinking itself. Its distinctive constructs—variables, causal links, feedback loops, stocks, flows, delays, boundaries, and related higher-order structures—provide a powerful language for representing and studying dynamic behavior. DSRP-484 makes a more elemental onto-epistemic claim: organization itself is generated through four recursively enclosed Patterns—Distinctions, Systems, Relationships, and Perspectives—composed of eight elemental roles and governed by four structural dynamics. This paper asks whether the organizational structures distinctive to system dynamics require anything outside DSRP-484. Using the O-Theory distinction between information and organization, $X=I \cdot O$, we develop a structural-remainder test that decomposes SD constructs until either their organization is fully generated by DSRP-484 or an irreducible residual remains. Variables decompose as distinguished identities; stocks as quantified part-whole Systems; flows as quantified Relationships that change Systems; feedback loops as Systems of recursively linked Relationships; delays as temporal information carried by Relationships or as decomposable process-Systems; boundaries as recursively enclosed DSRP structures; and aggregation and disaggregation as recursive part-whole transformations. Mathematical and computational machinery used by SD—including differential equations, numerical integration, and simulation—is not distinctive SD ontology and therefore does not enlarge the organizational basis supplied by SD itself. We then apply the structural-remainder test to a pre-committed candidate set of 60 constructs drawn from the field's own vocabulary under a fixed extraction rule. All 60 decompose without requiring an additional organizational primitive. No residual X is identified. The result supports strict structural containment: $O_{SD} \subset \text{Cl}(O_{\text{DSRP-484}})$. The result also has a methodological implication. If SD's characteristic constructs are specialized composites rather than organizational primitives, beginning inquiry with stocks, flows, feedback, and delays can bias the analyst toward seeing the organization those forms are designed to represent. DSRP can instead operate upstream: first determining what distinctions, systems, relationships, and perspectives the phenomenon contains, and only then determining which of those structures should be instantiated in an SD model—a workflow already demonstrated by Caddell and Nilchiani in their construction of a system dynamics model of interpersonal trust. System dynamics therefore remains a powerful domain-specific modeling method, but both its organizational foundations and the thinking used to construct its models can be situated within a more elemental organizational calculus.

KEYWORDS: DSRP | DSRP-484 | System Dynamics | O-Theory | Unified Calculus | Structural Containment | Organizational Ontology | Structural Decomposition | Stocks and Flows | Feedback

Cite this paper: Cabrera, D. (2026). Is system dynamics made of DSRP? Testing whether the organizational structure of system dynamics is contained within DSRP-484. *Journal of Systems Thinking*, 6(1), 1–59. <https://doi.org/10.54120/jost.v26i1.dc6sdos6>

1. The Question Beneath the Methods

The field of systems thinking proposes many methods, but they are not equivalent in their scientific substance or practical utility. Some function primarily as interpretive, normative, or ideological or political frameworks whose claims are difficult to operationalize or falsify; others provide explicit structures, measurable constructs, formal rules, and testable machinery.

Some traditions consist largely of conceptual vocabularies, interpretive frameworks, normative commitments, methodological prescriptions, or notoriety echo chambers (e.g., Critical Systems Thinking, Critical Systems Heuristics, Soft Systems Methodology, the Viable System Model, Systems Innovation, General Systems Theory, and Cynefin) [1]. Their lack of empirical constraint can become a methodological liability: central constructs are often difficult to operationalize, measure, or falsify, allowing the opinions and biases of their founders and proponents to become embedded in the framework itself. When a framework can accommodate nearly any observation after the fact—including observations congenial to the normative or political commitments of its founders—it becomes difficult to specify what evidence would show that the framework is wrong. At that point, practitioners face a basic question: what distinguishes the "method" from a formalized collection of its proponents' opinions and biases?

Other traditions provide considerably more explicit and testable machinery (e.g., system dynamics, network theory, complex adaptive systems, UML, hierarchical modeling, some systems engineering, and process modeling). Network analysis specifies nodes, edges, and graph structures; complexity and complex adaptive systems research formalizes interacting agents and emergent dynamics; UML specifies classes, objects, relationships, and transformations; process models specify states, transitions, inputs, and outputs; and hierarchical methods specify levels, parts, and wholes. System dynamics belongs firmly in this latter group. Its stocks, flows, feedback loops, delays, causal links, and variables can be explicitly specified and combined with mathematical and computational machinery to generate behavior that can be measured, simulated, and tested [2–5]. Yet these formal vocabularies differ sufficiently that they are often treated as fundamentally different kinds of systems machinery.

But a prior question can be asked: *What are these constructs themselves made of?* In Bateson's terms, "What is the pattern that connects?" More specifically: *What are the universal primitive elementals that underlie organization itself?*

Four terms require distinction at the outset. By *primitive*, we mean a structure belonging to the original generative basis of an organizational calculus: something from which other structures can be constructed. By *elemental*, we mean structurally irreducible: it cannot be decomposed into a more basic kind of organization. A *primitive elemental* therefore has both properties: it belongs to the generative basis and is not itself organizationally decomposable. By *universal*, we mean necessary to organization as such:

organization cannot exist without it. By contrast, a *composite* is any structure generated through the combination or recursion of primitive elementals. Higher-order composites may be scientifically indispensable, formally sophisticated, and domain-defining without thereby being elemental.

primitive elementals $\rightarrow$ composites $\rightarrow$ higher-order composites

A feedback loop is clearly an important structure. But is *feedback loop* a primitive elemental of organization, or is it a composite arrangement of simpler structures? The same question applies to a stock, a flow, a delay, and even a variable. *A construct can be foundational to a particular modeling tradition—and can even be treated as undecomposed within that method—while remaining organizationally composite.*

That distinction motivates the present paper.

System dynamics is a particularly strong test case because it is not merely a loose or poorly defined systems-thinking heuristic (like so many are). From Forrester's early work through contemporary SD practice, the tradition includes explicit causal structure, accumulation, rates, delay, nonlinear relations, differential or difference equations, and computer simulation [2,4]. Any general theory of organization that claims sufficiency should therefore be able to account not merely for generic notions such as "interconnection" or "wholeness," but for the actual representational machinery used by SD.

DSRP-484 makes precisely such a sufficiency claim. DSRP is formally specified as four Patterns—Distinction, System, Relationship, and Perspective—containing eight elemental roles: identity/other, part/whole, action/reaction, and point/view. These structures operate through Equality, Coimplication, Simultaneity, and Enclosure [6–9]. Under Enclosure, no Pattern is structurally independent: any instantiated Distinction requires System, Relationship, and Perspective; any System requires Distinction, Relationship, and Perspective; and so forth [7]. DSRP-484 is therefore not proposed as four descriptive categories but as a recursively generative structural calculus. In other words, DSRP-484 is not "4 buckets defined by the dictionary meaning of the four words." It is a highly defined calculus that makes numerous falsifiable predictions. The 4 Patterns, 8 elements, their special relationships, the 4 dynamics, and 6 emergent mechanics are definitional to this calculus and are well documented [6–9] and tested [10].

The question is consequently not whether DSRP can *label* the constructs of system dynamics. Any sufficiently vague taxonomy could do that. The stronger question is whether the organizational structure required by each SD construct can be decomposed into DSRP-484 *without leaving a residual organizational primitive.*

If so, the relation between the two theories is not simply that they are "alike":

$$DSRP \sim SD$$

or that one might "use both together":

$$DSRP + SD$$

but potentially that the organization of system dynamics (O_{SD}) is contained within the closure generated by $O_{DSRP-484}$:

$$O_{SD} \subseteq Cl(O_{DSRP-484})$$

where (Cl) denotes the structures generated through composition and recursion of DSRP-484.

In plain language:

System dynamics may be a domain-specific composite calculus whose organizational structures are composed entirely from the universal primitive elementals specified by DSRP-484.

This paper develops and places that claim at risk.

2. Information Is Not Organization

The argument requires a distinction between *information* and *organization*.

O-Theory expresses any organized phenomenon generally as:

$$X = I \cdot O$$

where I denotes informational content and O denotes its organization [6,8]. For cognition, where $\mathbb{M}$ is the meaning or mental model:

$$\mathbb{M} = I_{\mathbb{M}} \cdot O_{\mathbb{M}}$$

and for reality:

$$\mathbb{R} = I_{\mathbb{R}} \cdot O_{\mathbb{R}}.$$

The current claim concerns principally the O term. In O-Theory, organization is the structure into which information is instantiated. An analogy makes the distinction clear: $A:B::C:D$ specifies a relational structure, while the particular values substituted for a,b,c, and d supply its informational content. The structure can recur across indefinitely many different contents. The same distinction applies to system dynamics. Numbers, units, parameter values, time indexes, variable names, equations, and algorithms contain information that may be highly specialized, domain-specific, and diverse, while the ways in which those contents are distinguished, grouped, related, and framed constitute their organization. Mathematical and computational procedures are not exempt from this claim; they too possess organization. The question examined here is therefore not whether every piece of SD information is itself a DSRP Pattern, but whether the organization structuring that information requires any universal primitive elemental beyond DSRP-484. *Organization* is comparatively invariant and recursively reusable; *information* is comparatively variable,

specialized, and specific. The same *organizational* structure can be instantiated by wildly different *information*.

The relevant question is therefore narrower: when informational values and mathematical operations are organized into an SD model, does that organization require anything outside DSRP-484?

For any system dynamics construct c , O-Theory separates its informational content from its organization:

$$c = I_c \cdot O_c$$

Read plainly: a construct consists of some particular information, I_c , instantiated in some particular organization, O_c . The information may vary enormously from case to case; the present test concerns the organization.

We then ask whether the organization of that construct can be generated from DSRP-484:

$$O_c \in \text{Cl}(O_{\text{DSRP-484}})$$

Here, Cl denotes the *closure* of DSRP-484: all organizational structures that can be generated through composition and recursion of its primitive elementals. The symbol $\in$ means "is a member of." Thus the equation reads: the organization of construct c is one of the organizational structures that DSRP-484 can generate.

The containment claim therefore fails if some necessary organizational feature of an SD construct requires a primitive elemental X that lies outside that closure:

$$X \notin \text{Cl}(O_{\text{DSRP-484}})$$

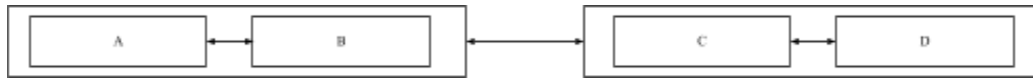
Thus, if some indispensable organizational structure in system dynamics cannot be reduced to, or generated through composition and recursion of, DSRP-484, then the containment hypothesis is false.

This also clarifies what is and is not being tested. The question is not whether DSRP "contains" arithmetic, calculus, numerical values, units, or particular simulation algorithms. Those are forms of information or specialized procedures instantiated within organized structures. The relevant question is whether the organization of those structures requires any *universal primitive elemental* beyond DSRP-484. Mathematical and computational procedures are themselves organized; they are not exempt from the theory. But their specialized informational content is not what the structural-remainder test is designed to classify.

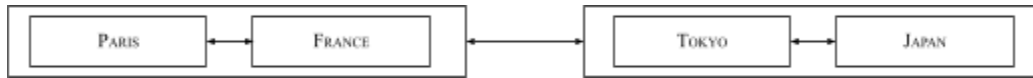
To make that distinction operational, we pre-commit to a decision rule before examining any SD construct. A feature is *organizational* if changing it changes what entities or relationships exist, or how they are distinguished, bounded, nested, ordered, related, or framed. A feature is *informational* if changing it alters the content or value instantiated in an organizational structure while leaving that structure intact. Consider again the generic symbolic logic of an analogy:

$$A : B :: C : D$$

or structurally:



Its *organization* is unchanged regardless of what *information* fills those positions. Thus,



and



instantiate different information within the same organizational structure: one thing is related to another in the same way that a third thing is related to a fourth. *Changing what fills the structure changes information; changing the structure itself changes organization.*

Where this paper names a concept as information filling a structure rather than as the structure itself, it is set in SMALL CAPITALS. PARIS and TOKYO above are *information*; the analogical structure that holds them is *organization*.

The rule works in both directions. Link polarity, for example, is informational: changing a positive sign to a negative sign changes the value carried by an existing relationship while leaving the relationship itself in place. The direction of a causal link, by contrast, is organizational: changing which relatum acts and which reacts changes the relationship that exists (Section 9). The distinction therefore cannot simply be used to classify inconvenient cases as "information." It can, in principle, force the discovery of a residual primitive elemental outside DSRP-484.

Applied across the full candidate set, the test becomes:

$$O_{SD} \subseteq Cl(O_{DSRP-484})$$

Or, in plain language: the organizational structures required by system dynamics are contained within the set of organizational structures that can be generated by DSRP-484. One legitimate counterexample is sufficient to break that containment relation.

The hypothesis is stated as containment rather than strict containment: the test asks only whether SD's organization falls inside the DSRP-484 closure, not whether it exhausts it. Section 17 shows the surviving result is the stronger one.

3. The Structural Remainder Test

We begin with the familiar ten-construct core of system dynamics:

$$C_{SD} = \left\{ \begin{array}{llll} \text{variable,} & \text{causal link,} & \text{feedback loop,} & \text{stock,} \\ \text{flow,} & \text{delay,} & \text{boundary,} & \text{aggregation,} \\ \text{rate,} & \text{simulation} & & \end{array} \right\}$$

These constructs are well established in the SD tradition [2–5]. Stocks represent accumulated states; flows change those stocks; causal relations connect variables; closed chains of causal relations form feedback loops; delays separate causes and consequences in time; and equations permit the resulting structures to be simulated [2,4].

This ten-construct inventory forms the canonical core treated in Sections 4 through 13. The full candidate set, C_{SD} , is defined by the pre-registered extraction rule in Section 15, which draws sixty constructs from the field's own vocabulary. The ten above are its most load-bearing members; the remainder are addressed in Sections 15 and 16.

For each construct c in that candidate set,

$$c \in C_{SD}$$

the analysis asks three questions:

1. What informational content is specific to this construct?
2. What organizational structure is necessary for the construct to exist?
3. Does that organization require a primitive elemental outside DSRP-484?

As established in Section 2, the test evaluates the organization an SD construct requires, not the mathematical or computational machinery used to quantify it. A numerical integration scheme is not a candidate construct; the state variables and stock-flow transitions its equations operate on are.

The containment hypothesis predicts that the organization of every SD construct will fall within the closure generated by DSRP-484:

$$H_A : \quad \forall c \in C_{SD}, O_c \in \text{Cl}(O_{DSRP-484})$$

In other words, however specialized an SD construct may be, its necessary organizational structure should be generable through composition and recursion of the primitive elementals specified by DSRP-484.

The competing hypothesis requires only a single exception:

$$H_0 : \quad \exists c \in C_{SD} \text{ such that } O_c \notin \text{Cl}(O_{DSRP-484})$$

The falsifier is unusually simple. If even one legitimate SD construct requires an indispensable organizational structure that falls outside that closure, the containment claim fails.

This is the structural remainder test. Each construct is decomposed until either its organization is fully accounted for by DSRP-484 or something organizational remains. If such a remainder cannot itself be generated through further composition or recursion, it constitutes the counterexample: a primitive elemental not contained within DSRP-484.

4. Variables Are Distinguished States

A system dynamics model begins by identifying variables. Before a model can say that A affects B , it must establish A and B as identifiable things. In practice, however, SD usually makes only the identity side of that distinction explicit:

$$D_i(A) = i_A$$

The corresponding and *coimplied* other,

$$D_o(A) = o_A$$

is often left implicit, ignored, or externalized beyond the immediate representation. Naming "population," for example, explicitly establishes an identity, but the model may never enumerate everything that counts as "not population." Likewise, "inventory," "demand," "price," "employee morale," and "pollution" are typically represented by naming what the variable is, not by explicitly representing everything it is not. The D_o structure exists, and is predictable, even when it is ignored, implicit, or externalized.

DSRP-484 does not therefore reduce Distinction to identity. Rather, every identified i_A coimplies an o_A , and the two are organized through the special relational structure of distinction: the identity *exists* as what it is, while the other *negates* that identity as what it is not. Thus the fuller structure is:

$$D_A = \left(i_A \begin{array}{c} \text{exists} \\ \rightleftharpoons \\ \text{negates} \end{array} o_A \right)$$

The abbreviated form,

$$D_A = (i_A, o_A)$$

therefore suppresses, but does not eliminate, the relational structure between identity and other. The important distinction is between what is structurally required and what is representationally explicit. SD commonly externalizes the i side while leaving the o side and the exists-negates relation tacit. DSRP-484 treats the full D_{io} structure as constitutive of a Distinction whether or not every element is explicitly represented.

The same variable is also situated as a part of the larger model-system:

$$p_A \in w_{model}$$

and the decision to treat that particular identity as a relevant variable is made from a perspective:

$$P = (\dot{p}, v)$$

Thus even the apparently simple SD variable is not structurally atomic. What SD typically externalizes as a named variable is principally the D_i side of a larger recursively enclosed structure. Its coimplied D_o , the exists-negates relation between them, its membership in a model-system, and the perspective under which it is selected may remain implicit, ignored, or externalized.

Its numerical value is information. Its organizational existence is a recursively enclosed DSRP structure.

This implication scales beyond the simple identity-other pair. Every distinguished object in a model has its own boundary, not only the model as a whole. If the model contains n distinguished identities, DSRP-484 predicts n corresponding others:

$$D_k = \{i_k, \mathfrak{r}_{io}, o_k\}, \quad k = 1, \dots, n$$

Thus:

$$n_i \rightarrow n_o$$

This becomes more powerful in the advanced Is/Is Not operation of contextual identity [11]. The identity of an object is reconstructed by assembling and counting the structurally true statements in which it participates: what it is and is not, what it is and is not related to, what wholes it belongs to or contains, and what it frames or is framed by. Contextual identity therefore captures not merely the label assigned to X but its complete structural position within the surrounding organization. Missing Relationships, memberships, or framings can themselves become analytically meaningful and can generate predictions about structures that may exist but have not yet been represented.

5. Causal Links Are Relationship-Systems

A causal link in a causal-loop diagram is normally represented as:

$$A \rightarrow B$$

DSRP-484 exposes the underlying organization of that arrow. Let A and B be the relata—the things being related—and let R_{AB} denote the Relationship in which A acts on B :

$$A \xrightarrow{R_{AB}} B$$

The relata are not themselves the action (a) and reaction (r) elements of the Relationship. Rather, R_{AB} is an object with its own internal organization. The action element a_{AB} is jointly contained by A and R_{AB} (*propagative containment*), while the reaction element r_{AB} is jointly contained by B and R_{AB} . Within the Relationship, those two elements are connected through the special relationship ($\mathfrak{r}$) *affects* and *effects*:

$$R_{AB} = \left(a_{AB} \begin{array}{c} \text{affects} \\ \rightleftharpoons \\ \text{effects} \end{array} r_{AB} \right)$$

Thus A and B are the relata, while a_{AB} and r_{AB} are the primitive relational elements associated with them inside R_{AB} . The a/r organization establishes the orientation of the Relationship: a_{AB} is associated with A acting on B , while r_{AB} is associated with B reacting to or being affected by A . The visible arrow is simply a representation of that underlying organization.

The *affects/effects* reciprocity internal to R_{AB} should not be confused with a second directional Relationship from B back to A . A single Relationship contains one action-reaction pair and their special relationship ($\mathfrak{r}_{ar}$):

$$R_{AB} = \{a_{AB}, \mathfrak{r}_{ar}, r_{AB}\}$$

and the special relationship connecting that pair do not create a second R . They specify the internal organization of the one Relationship.

Nor does this reciprocity require equality or symmetry. DSRP-484 requires the coimplicated action-reaction structure, but places no general constraint on the relative magnitude or form of a and r . In Newton's third law, for example, the paired effects are constrained:

$$a_{AB} = -r_{AB}$$

Newton's third law can be understood as a special physical case of reciprocal action-reaction organization in which the paired effects are constrained to be equal in magnitude and opposite in direction.

A distinct Relationship from B to A requires a second action-reaction pair:

$$B \xrightarrow{R_{BA}} A$$

with:

$$R_{BA} = \left(a_{BA} \begin{array}{c} \text{affects} \\ \rightleftharpoons \\ \text{effects} \end{array} r_{BA} \right)$$

Here a_{BA} is jointly contained by B and R_{BA} , while r_{BA} is jointly contained by A and R_{BA} . Therefore:

$$R_{AB} \neq R_{BA}$$

The same relata participate in both Relationships, but each Relationship has its own action-reaction pair. R_{AB} contains a_{AB} and r_{AB} ; R_{BA} contains a_{BA} and r_{BA} .

The important point is therefore not merely that the SD arrow can be relabeled R . The arrow compresses an internally organized relational object. To assert:

$$A \xrightarrow{R_{AB}} B$$

requires Distinctions establishing A and B ; a System in which A , B , and R_{AB} occur as parts; the action-reaction organization internal to R_{AB} ; and a Perspective from which the directional relation is framed. Under DSRP-484 these are recursively enclosed aspects of the Relationship rather than independent additions to it.

Because a Relationship is itself an object, it can also be opened. An SD model may represent:

$$\text{workload} \rightarrow \text{error rate}$$

as a single Relationship, while a finer-resolution model may expose:

$$\text{workload} \rightarrow \text{fatigue} \rightarrow \text{attention loss} \rightarrow \text{error probability}$$

What appeared at one level as a single Relationship is revealed at another as a System composed of multiple relata and Relationships. The original relation has not disappeared; its internal organization has been exposed.

SD correctly insists that behavior emerges from causal structure. DSRP asks one level lower: how is causal structure itself organized?

6. Feedback Loops Are Systems of Relationships

Feedback is foundational to system dynamics [2,4], but an SD feedback loop is not organizationally elemental. The key distinction is between the reciprocal organization internal to a single Relationship and the higher-order closure produced by multiple Relationships.

A single causal Relationship from A to B contains one action-reaction pair:

$$A \xrightarrow{R_{AB}} B$$

where:

$$R_{AB} = \left(\begin{array}{ccc} & \text{affects} & \\ a_{AB} & \xleftrightarrow{\quad} & r_{AB} \\ & \text{effects} & \end{array} \right)$$

This is one Relationship with two primitive relational elements, a_{AB} and r_{AB} . The reciprocal *affects/effects* structure between them belongs inside R_{AB} . It does not, by itself, constitute an SD feedback loop.

The minimal feedback loop requires a second Relationship in which B acts on A :

$$B \xrightarrow{R_{BA}} A$$

with its own action-reaction pair:

$$R_{BA} = \left(a_{BA} \begin{array}{c} \text{affects} \\ \xleftrightarrow{\quad} \\ \text{effects} \end{array} r_{BA} \right)$$

The minimal feedback loop therefore contains two distinct Relationships:

$$A \xrightarrow{R_{AB}} B \xrightarrow{R_{BA}} A$$

and two distinct action-reaction pairs:

$$R_{AB} = \{a_{AB}, \mathfrak{r}_{ar}, r_{AB}\}$$

$$R_{BA} = \{a_{BA}, \mathfrak{r}_{ar}, r_{BA}\}$$

Thus a single causal Relationship contains one a/r pair, or two instantiated relational elements, whereas the minimal SD feedback loop contains two a/r pairs, or four instantiated relational elements.

The count matters because a single directed Relationship ($\rightarrow$) contains two relational elements—one action and one reaction:

$$\{a_{AB}, r_{AB}\}$$

A minimal feedback loop contains two distinct Relationships ($\rightleftharpoons$) and therefore four relational elements:

$$\{a_{AB}, r_{AB}, a_{BA}, r_{BA}\}$$

That difference from two to four is consequential. Each action or reaction element is a dynamic locus at which effects can originate, be received, altered, amplified, dampened, redirected, or leveraged. A single Relationship therefore contains two such loci; the minimal feedback loop contains four, organized into two coupled action-reaction pairs. The additional dynamics of feedback arise not from a new primitive elemental, but from the recursive coupling of these four relational elements into a closed System.

The loop is therefore a higher-order System whose parts include the relata and the two Relationship objects:

$$S_L = \{A, B, R_{AB}, R_{BA}\}$$

Feedback adds no new primitive elemental. It arises when already-organized Relationship objects are composed into a System whose causal pathway closes back upon an earlier relatum.

The same logic scales recursively. Feedback can involve any number of relata and Relationships. For example:

$$A \xrightarrow{R_{AB}} B \xrightarrow{R_{BC}} C \xrightarrow{R_{CD}} D \xrightarrow{R_{DA}} A$$

forms a four-relatum loop containing four distinct Relationship objects:

$$S_L = \{A, B, C, D, R_{AB}, R_{BC}, R_{CD}, R_{DA}\}$$

Each Relationship contains one action-reaction pair, so this loop contains four *a/r* pairs and therefore eight instantiated relational elements:

$$\{a_{AB}, r_{AB}, a_{BC}, r_{BC}, a_{CD}, r_{CD}, a_{DA}, r_{DA}\}$$

It also contains four instantiations of the special action-reaction Relationship r_{ar} , one within each *R*. Thus, at the relational level, the four-link loop contains:

$$4 \text{ Rs} \rightarrow 4 \text{ a/r pairs} \rightarrow 8 \text{ relational elements}$$

For a simple closed loop containing *n* relata and *n* directed Relationships:

$$\text{feedback loop} = S(R_1, R_2, \dots, R_n)$$

the counts scale directly:

$$n \text{ Rs} \rightarrow n \text{ a/r pairs} \rightarrow 2n \text{ relational elements}$$

with *n* corresponding instantiations of r_{ar} . These counts matter because every additional *a* or *r* is another dynamic locus at which effects can originate, be received, altered, amplified, dampened, redirected, or leveraged. Increasing the size of the loop therefore increases the number of dynamically consequential relational elements without introducing any new primitive elemental.

What makes the structure an SD feedback loop is the closure of a causal pathway across multiple Relationship objects, not the internal reciprocity of any one Relationship. The latter is elemental to R_{ar} ; the former is a composite System built from multiple R_{ar} structures.

Reinforcing and balancing loops likewise require no additional primitive elemental. The positive or negative polarity assigned to an SD link is informational: it specifies how variation in the action is associated with variation in the reaction while leaving the underlying R_{ar} organization unchanged. Loop polarity is consequently a derived informational property of the polarities carried by its constituent Relationships. What distinguishes reinforcing from balancing feedback is therefore not a different kind of organization, but different information instantiated within the same composite relational structure.

Feedback remains analytically indispensable in SD while the feedback loop itself remains organizationally composite. Calling it composite does not diminish its importance; it identifies the organizational scale at which it occurs. Just as importantly, the decomposition exposes a core systems-thinking skill that SD often relies upon without stating explicitly: the ability to "count R." A learner who can identify how many distinct Relationships, action-reaction pairs, and relational pathways are present—and how they are composed into larger Systems—is not merely recognizing a loop diagram. They are perceiving the underlying relational organization from which the loop's dynamics emerge. In this sense, the feedback loop is the outcome; the constituent R_{ar} structures are the inputs from which that outcome is composed.

Loop dominance follows the same logic. When several loops act on the same stock, one loop's net effect may outweigh the others and determine the model's behavior mode; as values change, dominance can shift from one loop to another. The loops themselves are Systems of Relationships. Their link weights and polarities are information carried by those Relationships. "Dominates" is a comparative Relationship between loop-Systems, evaluated relative to a behavior of interest and therefore under a Perspective. A dominance shift is consequently a change in information that produces a change in organization: changing relational values alters which loop stands in the dominance relation. Rather than resisting decomposition, loop dominance shows how informational change and organizational change interact within recursively organized DSRP structure.

7. Stocks Are Systems That Persist Through Change

Stocks are accumulations and state variables in system dynamics [4,5]. Population, inventory, cash, atmospheric carbon, and workforce size are common examples. In SD, accumulation is often taught through the familiar bathtub metaphor: water accumulates in a tub while inflows add to it and outflows remove from it.



Figure 1. Inflow and Outflow and Bathtub Dynamics

At first pass, the organizational signature of a stock is part-whole structure. The accumulated water is a distinguished whole composed of parts:

$$S_{water} = \{p_1, p_2, \dots, p_n\}$$

Here S_{water} is the water-System, and $p_1 \dots p_n$ are whatever parts constitute that whole at the resolution being modeled—individual molecules need not literally be counted. What SD normally tracks is not the identity of each part but how much of the whole is present:

$$Q_{water}(t)$$

where Q denotes quantity and t indicates that the amount can change over time. This is the sense in which a stock can be generalized as:

$$\boxed{\text{stock} \approx \text{quantified } S_p^w}$$

The expression says that the defining first-order structure of a stock is a part-whole System carrying a quantity. The approximation is important. It identifies the primary organizational structure that makes a stock recognizable as a stock; it does not imply that an actual stock contains only S structure. Under DSRP-484, opening the bathtub reveals the recursively enclosed organization beneath that generalization.

The water is first distinguished as the water being counted rather than everything that is not that water. It is then organized as a System whose parts constitute a whole. The tub is another distinguished System, composed of its own parts, within which the water-System is contained:

$$S_{water} \in S_{tub}$$

The symbol $\in$ marks that containment: the water-System is functioning here as a part within the larger tub-System.

The water-System and tub-System also stand in a Relationship. The organization of the tub constrains the possible state of the water it contains. From the structural point established by the tub, only some views of the water-System are admissible: the water can occupy the available volume but cannot exceed the containing structure without becoming a different state of the larger system, such as overflow. The tub need not consciously "take a perspective"; its structure establishes a point from which the possible states of the contained water are framed. Remember that while perspective is a term commonly used anthropocentricly in DSRP-484 it is defined in terms of its framing function:

$$P := \left(v \begin{array}{c} \text{frames} \\ \rightleftharpoons \\ \text{framed} \end{array} \dot{p} \right)$$

Here the point $\dot{p}$ and view v form the framing structure: the organization of the tub establishes a point that frames what states of the water are physically available.

The numerical quantities involved remain information. If the tub has a capacity of 150 liters and currently contains 80 liters, then:

$$Q_{water} = 80$$

and:

$$Q_{capacity} = 150$$

The first number describes the present amount of water; the second describes the amount the containing structure can hold. Their Relationship produces the familiar constraint:

$$Q_{water} \leq Q_{capacity}$$

The water quantity may rise or fall, but while it remains contained by the tub it cannot exceed the quantity permitted by the tub's capacity. The numbers are *information*; the containing, constraining, and framing structure that makes the inequality meaningful is *organization*.

The bathtub therefore contains considerably more organization than the shorthand "stock = System" suggests. It includes a boundary Distinction, a part-whole System, a contained sub-System, Relationships between those Systems, and a structural Perspective that constrains or frames the admissible states of the contained whole.

DSRP-484 predicts this richer organization because no Pattern exists independently of the others.

The same logic applies to other stocks. For a population:

$$S_{\text{population}} = \{\text{person}_1, \text{person}_2, \dots, \text{person}_n\}$$

the population is the whole and the persons are its constituent parts. Its current size can then be represented as:

$$Q_{\text{population}}(t) = n$$

so n is simply the number of persons belonging to that population at time t .

For continuously measured stocks, the parts need not be individually enumerated. The important first-order fact is that some distinguished content is organized as a measurable whole. The quantity is information; the part-whole organization is what principally gives the construct its stockhood.

This also explains how a stock can persist while its contents change. A whole is not identical to its current parts. The bathtub remains the bathtub while water enters and leaves; a population remains the modeled population while persons enter and exit. The whole persists while its constituent parts and associated quantities change.

The population boundary distinguishes who counts and who does not; the persons constitute parts of the population whole; the population exists within larger containing Systems; Relationships govern entry, exit, and interaction; and the model's structural frame determines which population and which quantity are being considered.

Any stock can also be recursively decomposed through part-whole structure. "Workforce" may contain departments; departments contain teams; teams contain persons. "Inventory" may contain product categories, locations, lots, or individual units. In general, a whole can be opened to expose its parts:

$$w \rightarrow \{p_1, p_2, \dots, p_n\}$$

and recursion means that any one of those parts can itself be treated as another whole:

$$p_i \rightarrow w_i$$

A department, for example, may be a part of a workforce at one scale and a whole containing teams at the next.

The structural claim is consequently not that a stock "is only S ." It is that the characteristic organizational signature of stockhood is a quantified part-whole System, while the full organization of any instantiated stock unfolds recursively through DSRP-484.

8. Flows Are Relationships That Change Systems

Flows require the same distinction between a primary organizational signature and the richer structure revealed when the construct is opened. At first pass, if a stock is principally a quantified System, a flow is principally a quantified Relationship through which the informational state of that System changes.

For a stock S , the familiar stock-flow equation is:

$$\frac{dQ_S}{dt} = \sum q_{in}(t) - \sum q_{out}(t)$$

Here Q_S is the quantity contained in the stock, and dQ_S/dt is how quickly that quantity is changing. The stock rises when the combined inflows exceed the combined outflows and falls when the reverse is true.

Structurally, each inflow or outflow is a Relationship connecting the stock to something else. An inflow can be represented as:

$$X \xrightarrow{R_{XS}[q(t)]} S$$

The expression says that X stands in a directed Relationship R_{XS} to the stock S , while $q(t)$ records the quantity moving through that Relationship at time t .

An outflow reverses the orientation:

$$S \xrightarrow{R_{SY}[q(t)]} Y$$

Now the stock S is acting through a Relationship toward Y , again with $q(t)$ carrying information about the measured rate of transfer.

The Relationship supplies the primary organization of the transfer, while $q(t)$ supplies quantitative information about that Relationship. In this first-order sense:

$$\boxed{\text{flow} \approx \text{quantified } R}$$

A flow is therefore principally an action-reaction Relationship carrying a quantity or rate. Likewise, from the previous section:

$$\boxed{\text{stock} \approx \text{quantified } S}$$

The two expressions identify the characteristic organizational signatures of the constructs, not their exhaustive organization.

Again, the bathtub metaphor makes the richer structure visible. The water that flows is itself organized content—a System of parts. The pipe is also an organized object and therefore a System composed of parts. Yet, relative to the stock-flow process, the pipe realizes and constrains a Relationship through which the water-System is transferred.

The pipe can therefore be opened as a System:

$$S_{pipe} = \{p_1, p_2, \dots, p_n\}$$

where its parts may include the pipe wall, lumen, valve, fittings, and other structures. But in the larger stock-flow configuration that same pipe-System participates in realizing:

$$R_{flow}$$

The first expression asks what the pipe is made of; the second asks what relational role it plays in the larger process. The same object can participate in both structures because DSRP Patterns are recursively enclosed rather than mutually exclusive categories.

Its diameter, valve structure, length, geometry, and other parts constrain what can pass through it. From the structural point established by the pipe-System, only some flow states are admissible. A narrow pipe permits a different range of flows than a wide pipe; a closed valve permits a different range than an open valve.

The actual dimensions, valve position, maximum throughput, and observed rate are informational values instantiated within that organization. A simple quantitative constraint might therefore be written:

$$Q_{rate} \leq Q_{capacity}(R)$$

This does not introduce a new organizational primitive. The numerical limit is information. The constraining organization arises from the recursively interacting System, Relationship, and Perspective structures already present.

The transported water, the pipe, and the flow therefore should not be collapsed into one another. The water is a System of content being transferred; the pipe is a System whose structure realizes and constrains a pathway; the flow is the Relationship through which transfer occurs; and the measured rate is information carried by that Relationship.

The stock-flow equation can consequently be represented structurally as:

$$\frac{dQ_S}{dt} = \sum R_{in}[q] - \sum R_{out}[q]$$

Here $R_{in}[q]$ denotes incoming Relationships carrying quantified flow information and $R_{out}[q]$ denotes outgoing Relationships carrying the same kind of information. Their combined effect determines how the quantity associated with the stock-System changes.

The important result is therefore not that stocks "are S " and flows "are R " in an exhaustive sense. Those are their primary organizational signatures within SD. Once opened, both reveal recursively enclosed Distinctions, Systems, Relationships, and Perspectives carrying specialized information.

Stocks and flows are therefore not additional primitive elementals of organization. They are recognizable higher-order composites whose characteristic forms can be summarized as:

$$\text{stock} \approx \text{quantified } S_p^w$$

and:

$$\text{flow} \approx \text{quantified } R_{ar}$$

while their full organizational structures remain compositions generated within:

$$\text{Cl}(O_{DSRP-484})$$

That final expression places both constructs inside the closure of DSRP-484: however richly a particular stock or flow is instantiated, its organization can be generated through recursive composition of the DSRP-484 structures rather than requiring an additional primitive elemental.

9. Delays Are Time-Based Relationships

Delays are another central source of dynamic behavior in SD [4,5]. But "delay" can refer to two related structures.

"Delays Are Time-Based Relationships" means that a delay is a Relationship considered in terms of time, t . In the same way that "X in terms of Y" in mathematics recontextualizes X under a new frame, the temporal qualification does not make the Relationship something other than a Relationship. It places the Relationship under a temporal Perspective: from the point of time, what is the view of this Relationship? The lag, τ , is information carried within that framing.

The relationship can therefore be summarized as:

$$\boxed{\text{Delay} = R \text{ under } P_t \text{ carrying } I_\tau}$$

That is, the underlying organization is a Relationship, time supplies the Perspective under which that Relationship is considered, and the particular lag τ is information carried within that framing. At finer resolution, the apparent Relationship may itself open into a process-System of intermediate Relationships.

This is the temporal property of a Relationship:

$$A(t) \xrightarrow{R_\tau} B(t + \tau)$$

Here A acts through Relationship R at time t , while the corresponding state of B occurs after an interval τ . The symbol τ is simply the amount of lag. Changing τ changes when the effect occurs, not the underlying Relationship connecting A and B . Thus, delay is temporal information carried by R rather than a new organizational structure.

A caution is required here, because "temporal" can bundle two different features that should not be conflated: relational orientation and lag. Relational orientation specifies which relata occupy the action and reaction roles in a particular Relationship. Lag specifies when the corresponding change occurs. The two are independent.

A cause can precede its effect—smoking may produce cancer only much later—but causation can also be simultaneous. Kant's lead ball resting on a cushion causes the hollow at the very moment it rests there [12]. The Relationship is oriented even though:

$$\tau = 0$$

That is, a Relationship can have action-reaction organization without requiring any temporal separation between the two.

Newton's third-law force pair illustrates a different case. The interaction is simultaneous and bi-directional: both orientations are instantiated, and each force is both action and reaction within the reciprocal interaction because either force can occupy either relational role without changing the physical Relationship. Symmetry concerns the equivalence of those orientations; it is not the opposite of directionality.

Under the decision rule of Section 2, relational orientation and lag therefore classify differently. Orientation is organizational: changing which relata occupy the action-reaction roles changes the R_{ar} structure being instantiated. Bi-directionality occurs when both orientations are present. Lag is informational: changing the delay from THREE days to FIVE—or to ZERO—changes the value associated with the Relationship while leaving its relational organization intact.

The arrow in an SD model therefore represents relational orientation; it should not be read as meaning "later than." Temporal order can then arise when oriented Relationships are composed into chains—precisely the strategy by which the causal theory of time [13] and causal set theory in physics [14] construct temporal structure from causal structure rather than assuming temporal order first. The lag magnitude itself remains information attached to the Relationship.

This invites a question. If temporal order is generated by composing directed Relationships, what distinguishes temporal ordering from spatial or logical ordering, which compose in precisely the same way?

The compositional machinery is the same, and so is the Perspective through which the composition is read. A Perspective is a point and a view, and that organization does not change when the kind of order (information) changes. What changes is the information instantiated in the point. Fill the point with *TIME* and the composed chain yields sequence; fill it with *SPACE* and the same chain yields arrangement; fill it with *ENTAILMENT* and it yields logical consequence. The point of a Perspective can be filled by *anything*, and what fills it constrains what the view returns without altering the structure that returns it.

The difference between temporal, spatial, and logical order is therefore not a difference of *organization* at all. It is different *information* instantiated in the same organizational structure — the distinction of Section 2, applied one level up, to the framing rather than to the thing framed.

Or it can arise from above: a Perspective coimplies a view, and the view comes back ordered by the point that framed it. That ordering happens in three ways at once. It is *sequenced* — the point ranks what it frames, as the perspective of *TIME* ranks events. It is *filtered* — the point picks its own structure out of what was otherwise undifferentiated, as the perspective of *COLOR* picks the colors out of a painting and leaves the brushwork, the subject, and the composition unattended. And it is *differential* — the order returned is not the existing order or the order another point would have returned. Every Perspective does all three.

All routes are *organizational*; the point and view exist. What remains *informational* is what fills the point — which may be any concept whatever — and with it which order the view returns. This is also why lag classifies as it does: τ is a value carried on a Relationship whose orientation is already fixed and whose temporal framing — *TIME* being what fills the point of that Perspective — is already established. Changing τ alters neither.

The third route is what Section 18 relies on: what a population Perspective returns about an elk herd is not what a forage Perspective returns, and the difference is not a matter of detail

But delays frequently hide mechanism. What appears at one level as a single delayed Relationship:

$$A \xrightarrow{\text{delay}} B$$

may, when opened, turn out to contain a sequence of intermediate states and Relationships:

$$A \rightarrow p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow B.$$

Instead of one unexplained jump from A to B , the second expression exposes what happens in between. The apparent Relationship has been opened as a process-System whose parts are the intermediate states and the Relationships connecting them.

This gives a particularly useful analytical question: *What is inside the delay?*

Sometimes the correct modeling choice is to preserve the aggregate delay. Sometimes opening it reveals the mechanism responsible for the lag. DSRP does not force one resolution. It makes the choice explicit

as a part-whole and Perspective operation: what appears as one Relationship at one scale may become a System of Relationships at another.

A delay can therefore remain a lag value attached to a Relationship:

$$\tau(R)$$

meaning that Relationship R carries temporal information τ , or it can be opened at finer resolution as:

$$S(R_1, R_2, \dots, R_n)$$

where what looked like one delayed Relationship is now represented as a System composed of multiple Relationships.

In one case the delay is information on R ; in the other, the apparent delay conceals additional organization inside R . Neither requires an additional primitive elemental of organization.

10. Boundaries Are DSRP Structures at Every Scale

System dynamics gives considerable attention to model boundaries because model behavior depends on what is treated as endogenous and exogenous [4].

A boundary is often spoken of as though it were simply a line around a model. Structurally, however, a boundary is already a composite DSRP structure.

Most basically, a boundary makes a Distinction. Something is treated as inside and, by the same act, something else is treated as outside. In DSRP-484 the full Distinction is not merely an identity standing alone but an identity-other pair connected through its special Relationship:

$$D_B = \{i_B, \mathfrak{r}_{io}, o_B\}$$

The special Relationship between the two can be opened as:

$$i_B \begin{array}{c} \xrightarrow{\text{exists}} \\ \xleftarrow{\text{negates}} \end{array} o_B$$

The inside therefore "is" what belongs to the modeled identity, while the outside supplies the coimplied "is-not" against which that identity is formed. The boundary is not added after the identity has already been established; the identity and its other arise together.

For an SD model, this can be written simply as:

$$i_B = \text{inside}, \quad o_B = \text{outside}$$

The Relationship between them is what makes the boundary a boundary rather than two unrelated regions. Inside and outside mutually define the distinction being made.

The same boundary also establishes a System. The modeled whole contains selected parts:

$$w_B = \{p_1, p_2, \dots, p_n\}$$

Here w_B is the model-whole and the p_i are the variables, stocks, flows, feedback structures, and other components treated as belonging within it. Changing the boundary therefore changes not only what is outside but which parts constitute the modeled whole.

Perspective supplies the framing function that determines why this particular distinction and whole are relevant:

$$P = (\text{modeler's point, bounded view}).$$

There is no boundary independent of a criterion for inclusion, and no criterion independent of a framing purpose.

$$P_B := \left(v_B \begin{array}{c} \text{frames} \\ \xleftrightarrow{\quad} \\ \text{framed} \end{array} \dot{p}_B \right)$$

The point-view structure need not be understood anthropocentrically. Its function is framing: from a particular structural point, some things are framed as belonging to the modeled view and others are not. A model boundary therefore contains all four Patterns. Distinction establishes inside and outside; Relationship connects and co-defines those sides; System organizes the selected inside as a whole of parts; and Perspective frames the criterion under which that organization counts as the relevant model.

Thus:

$$\boxed{\text{boundary} \in \text{Cl}(D, R, S, P)}$$

or, more compactly:

$$\boxed{\text{boundary} = D + R + S + P}$$

The equation should not be read as four independent ingredients simply added together. Under DSRP-484 they are recursively enclosed aspects of the same organizational act.

This also exposes an important feature that ordinary SD notation tends to compress. Every distinguished identity has its own coimplied other. If a model contains many distinct objects, then each object has an identity-other structure even though the diagram may represent externality with only one or a small

number of cloud symbols. The graphical cloud therefore operates at the level of the model boundary: it reifies an outside of the modeled whole. It does not eliminate the many object-level others required to distinguish each stock, variable, flow, or other construct.

DSRP-484 consequently permits an explicit count that the diagram often suppresses. For each distinguished object:

$$D_k = \{i_k, \mathbf{r}_{io}, o_k\}$$

so a model containing n distinguished identities contains n corresponding identity-other distinctions, whether their o_k elements are explicit, implicit, ignored, or externalized. This requirement helps ensure that constructs are fully defined rather than relying on an unspecified background of "everything else."

The point also explains why identity is contextual rather than absolute. What something "is" depends partly on what is being treated as its relevant "is-not." A population may mean residents of a city, citizens of a country, persons currently present within a jurisdiction, or some other bounded whole. The label may remain "population," while changing the other against which it is distinguished changes the identity actually being modeled.

This decomposition explains why two competent SD modelers can place different boundaries around the same phenomenon without either making a simple factual error. They may distinguish different identities and others, construct different wholes from those distinctions, or frame the phenomenon from different point-view structures.

It is worth noting what this decomposition captures. Richardson has argued that the deep core of system dynamics is not stocks and flows at all but the endogenous point of view—the commitment to explaining behavior from within the model boundary [15]. That is the field's own most senior theorist identifying its crown jewel, and the jewel decomposes on sight: endogeneity is a distinction between inside and outside, organized as a whole of selected parts, related across a boundary, and established under a frame. The construct SD's own literature nominates as foundational is therefore itself a DSRP composite.

The model boundary is therefore not a unique kind of Distinction. It is one particularly important Distinction drawn at the scale of the modeled whole. DSRP-484 predicts the same identity-other structure recursively for every distinguished object inside that whole.

SD makes this model-level outside unusually visible through its cloud notation; Section 16.3 returns to why that is both conceptually important and structurally incomplete.

11. Differential Equations Add Quantification, Not New Organization

The strongest apparent challenge to the containment argument is mathematical. System dynamics does not merely diagram. It uses rate equations, differential equations, integration, nonlinear functions, and simulation [2,4].

This does not place mathematics outside DSRP-484. Mathematical expressions are themselves organized objects and, if DSRP-484 is universal, their organization must also be decomposable through Distinctions, Systems, Relationships, and Perspectives. Consider:

$$\frac{dX}{dt} = F(X, Y, Z)$$

As a mathematical object, this expression already contains organization. It contains distinguished objects such as d , X , t , F , Y , and Z ; part-whole Systems such as dX/dt and $F(X,Y,Z)$; Relationships including division, functional dependence, and equality; and Perspectives that frame, for example, change in X with respect to t and X,Y,Z as arguments of a function.¹

The point of the present analysis is therefore not that mathematics lacks DSRP organization. It is that the specialized mathematical machinery used by SD does not require an additional primitive elemental of organization. The distinction is one of level.

An equation can itself be treated as an organized phenomenon:

$$X_{math} = I_{math} \cdot O_{math}$$

where I_{math} is the specific mathematical content and O_{math} is the organization of that mathematical object. If DSRP-484 is universal, then:

$$O_{math} \in Cl(O_{DSRP-484})$$

That same mathematical object can then function as part of the information instantiated within a larger SD model:

$$X_{SD} = I_{SD} \cdot O_{SD}$$

In other words, an equation may be an organized object at one level while functioning as information within another organized object at a higher level. There is no contradiction. This is exactly the recursive relation between information and organization developed in Section 2. The same principle underlies human communication. Person A constructs meaning by organizing information; that meaning is then

¹ The decomposition can be pushed further. For example, dX/dt can be treated as a System containing distinguished components dX , dt , and the relational operator connecting them; $F(X,Y,Z)$ is another System containing a distinguished function and its arguments; and the equation as a whole is a larger System whose two sides stand in an equality Relationship. Each of these objects can itself be recursively opened under DSRP-484. The purpose here is only to establish that mathematical notation is organized, not to provide a complete DSRP decomposition of calculus.

encoded into language and transmitted across a communication channel. What crosses the channel is not Person A's mental organization itself but an informational signal. Person B must reconstruct organization from that information to produce meaning. If Person B applies sufficiently corresponding organization to sufficiently corresponding information, the resulting meanings may align; if the organization differs, the meaning may differ even when the transmitted information is identical. *Understanding is therefore reconstructed, not transmitted.* As the familiar quip puts it, "*You can explain things to people, but you cannot understand things to people.*" The awkward construction captures the structural point precisely: information can cross the communication channel, but understanding cannot. Understanding must be reconstructed by the receiver by organizing the incoming information.

Thus, the same object may function as organized information, $I \cdot O$, when opened at one level of analysis and as information, I , when instantiated within a larger organizational structure. Information and organization are therefore recursively role-dependent without ceasing to be objectively distinguishable.

The narrower question for this paper is therefore whether rate equations, differential equations, integration, nonlinear functions, or related mathematical forms introduce an additional primitive elemental into the organization of system dynamics. They do not. They specify, constrain, and quantify changes occurring within organizational structures already present in the model.

For example, the stock-flow equation:

$$\frac{dQ_S}{dt} = \sum q_{in}(t) - \sum q_{out}(t)$$

does not introduce a new kind of organizational structure. The stock S has already been decomposed as principally a quantified part-whole System, and the inflows and outflows as quantified Relationships. The differential equation specifies how the informational quantity associated with that System changes as a function of the informational quantities carried by those Relationships.

The equation can therefore be written structurally as:

$$\frac{dQ_S}{dt} = \sum R_{in}[q] - \sum R_{out}[q]$$

The left side quantifies change in the informational state of the stock-System. The right side aggregates the quantified effects carried by incoming and outgoing Relationships. Calculus supplies the mathematical machinery for describing that change precisely; it does not thereby introduce a fifth kind of organization.

Under O-Theory, any organized phenomenon can be represented generally as:

$$X = I \cdot O$$

Dynamic change can therefore involve change in information, organization, or both.

A common case in SD is change principally in informational state while the organization of the model remains fixed:

$$\Delta I \neq 0, \quad \Delta O = 0$$

A population may increase from ten thousand to twelve thousand while the same stock, flows, causal Relationships, boundary, and framing structure remain in place. The model's information changes while its organization does not.

Change can also occur in organization:

$$\Delta O \neq 0$$

A new causal Relationship may appear, an existing Relationship may disappear, a model boundary may shift, a stock may be disaggregated into several stocks, a subsystem may be reorganized, or the network of causal dependence may change. These are not merely changes in values. They alter what entities or Relationships exist, how they are bounded or nested, or how they are framed.

Information and organization can also change together:

$$\Delta I \neq 0, \quad \Delta O \neq 0$$

Indeed, informational change can sometimes produce organizational change at a higher level:

$$\Delta I \rightarrow \Delta O$$

Loop dominance provides one example. The underlying Relationships and their polarities may remain in place while changing parameter values alter which loop stands in the dominance Relationship relative to the behavior of interest. The informational state changes first; the higher-order organization subsequently changes.

This distinction helps clarify what differential equations ordinarily do in SD. Most conventional simulation holds the model's organization relatively fixed while allowing informational states to evolve through time. In that common case, the equation describes:

$$O_t = O_{t+1}$$

while:

$$I_t \neq I_{t+1}$$

The organized structure persists while the values carried by that structure change.

But this is not a limitation of DSRP-484. It is a modeling choice. A model may also represent organizational transformation, in which case:

$$O_t \neq O_{t+1}$$

and the change itself becomes an object of analysis.

A general state transition can therefore be expressed as:

$$X_t = (I_t, O_t) \rightarrow X_{t+1} = (I_{t+1}, O_{t+1})$$

This representation makes explicit the question that ordinary state equations often leave tacit: what, precisely, changed? The information, the organization, or both?

Where I and O have been explicitly encoded as differentiable numerical quantities and the product form $X=I \cdot O$ is intended literally, the familiar product rule can also be applied:

$$\frac{dX}{dt} = O \frac{dI}{dt} + I \frac{dO}{dt}$$

But the containment argument does not depend on treating every occurrence of $I \cdot O$ as scalar multiplication. The more general proposition is that dynamic equations quantify transformations of informational states, organizational states, or both.

This also clarifies the status of mathematical operations such as integration. Integration is itself an organized mathematical operation and therefore falls within the same universal claim. At the level of an SD model, however, numerical or analytical integration functions as machinery for recovering changing informational states from specified rates of change. Its use does not add an additional primitive elemental to the organization of the modeled system.

The same logic applies to nonlinear functions. A nonlinear mapping is mathematically organized, but its nonlinearity does not by itself create a new organizational Pattern. If:

$$Y = F(X)$$

changes from a linear to a nonlinear functional form while the same $X \rightarrow Y$ Relationship remains, the mapping carried by that Relationship has changed while the basic relational organization has not. If the altered function causes new Relationships, boundaries, or higher-order organizational structures to appear,

then the consequence includes ΔO . The mathematical form and the organizational consequences should therefore be distinguished rather than conflated.

Differential equations thus add mathematical precision, expressive power, and computational tractability to system dynamics, but they do not identify an additional primitive elemental of organization.

DSRP-484 does not compete with calculus at the same level of analysis. Calculus supplies formal operations for quantifying change. DSRP-484 specifies the organization of the mathematical objects themselves and of the modeled structures to which those operations are applied.

12. Simulation Computes Organized Change, Not New Organization

A simulation model repeatedly advances a model from one state to the next:

$$X_t \rightarrow X_{t+\Delta t}$$

Here X_t denotes the model state at the current time t , Δt is one simulation time step, and $X_{t+\Delta t}$ is the state after that step. The arrow therefore means simply that the simulation advances the model from its current state to its next state.

Under O-Theory, the state at time t can be written as:

$$X_t = I_t \cdot O_t$$

where I_t is the information carried by the model at that time and O_t is the organization of that information. Simulation can therefore be represented as repeated transformation:

$$(I_t, O_t) \rightarrow (I_{t+\Delta t}, O_{t+\Delta t})$$

The expression says that each simulation step maps the model's current information and organization to the information and organization present at the next time step, according to the equations and constraints specified by the model. In most conventional SD simulations, the organization remains fixed while the informational values change, but the more general expression allows either or both to change.

Numerical integration determines how that transition is computationally approximated. The model's equations determine the quantitative rules. DSRP structure specifies how the modeled entities, boundaries, part-whole structures, Relationships, and Perspectives are organized.

Simulation therefore does not stand in opposition to DSRP. It operates over organized structures.

The numerical solver is computational machinery operating on those organized objects. It too is organized, but its organization belongs to the modeling procedure rather than to the represented system.

That distinction deserves one further defense, because this paper treats order as organizational—that is how temporal order was generated in Section 9—and a numerical integrator is nothing if not ordered steps. The answer is not that procedural order fails to be organization. Under O-Theory the analyst is an organized target too (Section 2) [8], and the solver's organization lives in the modeling act rather than in the model's claim about the world: its steps are composed directed Relationships, its state variables are Distinctions carried by a System, its timestep is information, and the choice of method is a Perspective. The exclusion is a scoping ruling, not an immunity ruling: the candidate set tests the organization an SD model attributes to the system it represents; the model claims nothing about the sequence in which a computer visits its equations, and mind-side organization has its own tests [16,17]. Nor does the ruling shelter the claim: a critic who forces the integrator into the candidate set finds not a residual but another decomposition.

13. The Canonical Constructs Share One Organizational Grammar

Table 1. DSRP-484 Decomposition of Canonical System Dynamics Constructs. Canonical system dynamics constructs decomposed into their underlying DSRP-484 organization. The table distinguishes each construct's primary SD function from the organizational structure required for it to exist and records whether any additional organizational primitive is required. Across the constructs shown, none is identified.

System dynamics construct	Primary SD function	DSRP-484 decomposition	Additional organizational primitive required?
Variable	Named measurable state	Distinguished identity (D), situated in (S), under (P)	None identified
Causal link	Specifies influence	Action-reaction Relationship (R)	None identified
Feedback loop	Circular causality	System (S) composed of recursively linked (R)'s	None identified
Reinforcing/ balancing polarity	Direction of loop effect	Information attached to relational configuration	None identified
Stock	Accumulation/state	Quantified part-whole System (S)	None identified
Flow	Rate adding/removing stock	Quantified relational transformation (R)	None identified
Delay	Temporal separation	Temporal information on (R), or recursive process-(S)	None identified
Boundary	Defines endogenous model	Boundary — recursive D+R+S+P structure; model-level case of universal identity-other boundary formation	None identified

Aggregation	Combines lower-level states	Recursive S_p^w	None identified
Disaggregation	Opens aggregate structure	Recursive $w \rightarrow \{p_i\}$	None identified
Rate equation	Quantifies state change	Quantification of ΔI , ΔO , or both	Mathematical operator, not new (O)
Differential equation	Continuous dynamic specification	Quantified transformation over DSRP-organized variables	Mathematical operator, not new (O)
Numerical simulation	Computes behavior over time	Repeated transformation $(I_t, O_t) \rightarrow (I_{t+1}, O_{t+1})$	Computational procedure, not new (O)
Loop dominance	Determines which loop governs behavior mode	Comparative (R) between loop-(S)'s, computed from weighted information, under (P)	None identified (shift = $\Delta I \rightarrow \Delta O$ coupling)

The result is not that every SD construct corresponds one-to-one with one DSRP Pattern. That would be the wrong theory.

The result is that the constructs are *composites* generated through DSRP organization.

14. One Model, Fully Decomposed: Population Growth

Consider the elementary SD population model:

$$\frac{dP}{dt} = B - D$$

Here P is population, B is births, and D is deaths. The equation says that population changes according to births entering the stock minus deaths leaving it.

Suppose:

$$B = \beta P$$

and:

$$D = \delta P$$

where β and δ are the birth and death rates per member of the population. Substitution gives:

$$\frac{dP}{dt} = (\beta - \delta)P$$

Thus the population grows when $\beta > \delta$, declines when $\delta > \beta$, and remains unchanged when the two rates are equal.

The same model can now be decomposed organizationally in Table 2:

Table 2. DSRP-484 Decomposition of an Elementary Population-Growth Model. Organizational decomposition of the elementary population-growth model developed in Section 14. The table shows how the population, stock, birth and death flows, feedback structures, rate equation, and model Perspective correspond to DSRP-484 structures while retaining their ordinary system dynamics functions.

SD construct	Representation	DSRP-484 decomposition	Meaning
Population	P	Distinction + System	The population is distinguished from non-population and treated as a whole composed of members.
Population stock	$S_P = \{\text{population members}\}$	Part-whole System	The population persists as a whole while its members and quantity change.
Birth flow	$B = \beta P$	Quantified Relationship	Birth is a Relationship that adds members to the population; βP supplies the quantity carried by that Relationship.
Death flow	$D = \delta P$	Quantified Relationship	Death is a Relationship that removes members from the population; δP supplies the quantity carried by that Relationship.
Birth feedback	$P \rightarrow B \rightarrow P$	System of Relationships	Population affects births, and births in turn increase population.
Death feedback	$P \rightarrow D \rightarrow P$	System of Relationships	Population affects deaths, and deaths in turn decrease population.
Rate equation	$\frac{dP}{dt} = B - D$	Quantified change over organized structures	The equation specifies how the quantity associated with the population-System changes through the two flow-Relationships.
Perspective	Population, place, time, membership criteria	Perspective	The model is framed around a particular population, time period, spatial boundary, and definition of membership.

Nothing is removed from the SD model by this decomposition. The stock, flows, feedback loops, rates, and differential equation remain exactly where they were. What changes is the level of explanation: the

familiar SD constructs can themselves be opened to reveal the more elemental DSRP organization from which they are composed.

15. One Residual Would Falsify the Claim

A universality claim has scientific value only when failure is clear.

The structural-containment hypothesis would be falsified by an uncontested construct necessary to system dynamics whose organizational structure cannot be represented through any composition or recursion of:

$$\{D, S, R, P\}$$

These are the four DSRP Patterns—Distinctions, Systems, Relationships, and Perspectives—with their eight elemental roles, special relationships, four dynamics and six mechanics, etc., as defined in the literature [6–9].

Formally, the claim fails if there is even one legitimate SD construct c whose organization O_c falls outside the space generated by DSRP-484:

$$\exists c \in C_{SD} \text{ such that } O_c \notin \text{Cl}(O_{DSRP-484})$$

Here C_{SD} is the candidate set of system dynamics constructs, O_c is the organization required by a particular construct, and $\text{Cl}(O_{DSRP-484})$ is the full space of organization that DSRP-484 can generate. The symbol $\exists$ means "there exists at least one." Thus, the equation says simply: if even one necessary SD construct requires organization that DSRP-484 cannot generate, the containment claim is false.

A legitimate failure could therefore be a construct requiring a fifth irreducible Pattern, a ninth elemental role, or an additional irreducible structural dynamic. Any such remainder can be represented simply as:

$$X \notin \text{Cl}(O_{DSRP-484})$$

where X denotes the organizational structure that remains after decomposition. If such an X exists and cannot itself be generated through further composition or recursion of DSRP-484, it is the surviving counterexample.

Mathematical and computational structures are not exempt from the universality claim of DSRP-484. If differential equations, numerical integration, stochastic sampling, algorithms, or any other mathematical or computational structure required an irreducible form of organization outside DSRP-484, that would constitute a legitimate counterexample to the broader universality claim. But it would not, by itself, constitute a counterexample to the present test, because those structures are not system dynamics. They are mathematical and computational machinery used by system dynamics and by many other fields.

The candidate set in this paper is therefore deliberately restricted to organization constitutive of SD itself. To treat every tool used by SD as part of the SD ontology would be a category error. Mounting a space

station on a Volkswagen Beetle would not make the space station part of the automobile's underlying design. Likewise, using calculus, differential equations, or numerical solvers within an SD model does not make calculus, differential equations, or numerical solvers distinctive organizational primitives of system dynamics.

The exclusion is therefore a scoping decision, not an immunity claim. Mathematics and computation remain fully exposed to the larger DSRP-484 universality claim; they are simply not what this particular structural-remainder test is testing.

One objection to that scoping decision deserves to be stated in its strongest form, because it is not the general complaint that mathematics has been excluded. It is specific, and it has support in the field's founding documents. Forrester did not treat continuous time as a computational convenience. He held that the world itself is continuous, that discrete-event representations distort it, and that continuous flows are the proper form for representing accumulation and change [2]. A critic may therefore argue that continuity is constitutive of system dynamics rather than borrowed by it, and that the extraction rule above, applied honestly to Forrester, should have placed it in the candidate set instead of filtering it as numerical procedure.

Three things answer that. The first is that Forrester's continuity commitment is a claim about the target, not about the model. "The world is continuous" is an assertion about reality, and DSRP-484 does not contest it. O-Theory is compatible with a continuous world and with a discrete one, because continuity concerns what is being organized rather than how it is organized.

The second is that no executed SD model is continuous. Simulation proceeds by numerical integration over finite time steps [4]. The continuous formulation is a limit toward which the difference equations are directed; the artifact that runs, generates behavior, and is compared against a reference mode is discrete. If continuity were constitutive of system dynamics, no model anyone has ever simulated would be a system dynamics model.

The third is structural. The continuous-discrete choice specifies the grain at which a temporal Perspective is taken. Section 9 establishes time as a Perspective under which a Relationship is considered; moving between continuous and discrete time changes the resolution of that Perspective—the information occupying the point—while leaving the point-view structure intact. The same identities are distinguished, the same wholes have the same parts, the same relata stand in the same Relationships, and the frame remains temporal. Under the decision rule of Section 2, that is an informational change, not an organizational one.

The falsifier remains available here as everywhere else, and it is worth saying what taking it would require. A critic who holds continuity to be organizational must exhibit a structure that continuity generates and that DSRP-484 cannot. Continuity does not generate structure. It specifies the grain at which a temporal Perspective is taken.

Nor is successful post hoc labeling sufficient to confirm the claim. A proper test must attempt decomposition prospectively and permit a residual category, *X*.

The test can therefore be stated very simply. For every construct in the candidate set, decomposition must end in one of two outcomes:

$$O_c \in \text{Cl}(O_{DSRP-484})$$

or:

$$X \notin \text{Cl}(O_{DSRP-484})$$

The first means that the construct's organization is fully generable from DSRP-484. The second means that something organizational remains that DSRP-484 cannot generate. The decisive result is therefore not a high percentage of matches. It is the existence or absence of X .

That test has now been constructed and executed in full, with construct-by-construct rulings recorded in the supplementary instrument. We pre-committed to an extraction rule before examining any construct: the candidate set C_{SD} is drawn exclusively from documents the SD field itself authored as statements of its vocabulary and foundations—every term defined in the field's official glossary [18], every construct named as a foundation or thread in Richardson's reflections [15], and the subject matter Forrester named for the field's second half-century, "high-order nonlinear feedback structures" [19]. An entry is excluded only under five pre-declared categories (numerical procedure, modeling practice, notation artifact, general term, imported theory), and every exclusion is logged with its category; nothing is dropped silently. Ford's glossary performs part of this filter in the field's own words, since its introduction excludes software-specific and general terms by design [18]. Cross-referenced entries are merged into canonical constructs, and each construct is then classified under the decision rule of Section 2. Nothing was added to the candidate pool. Sterman's textbook glossary [4] serves as a robustness check on the nominators: any construct defined there but absent from the nominating sources can be appended to C_{SD} under the same filter and ruled under the same criterion, without disturbing any existing ruling.

Applying the rule yields sixty canonical constructs and seventeen logged exclusions. Of the sixty, twenty-one are decomposed in Sections 4 through 13, twelve are composites of already-decomposed elements, three are purely informational, and twenty-four are decomposed in Section 16—among them the auxiliary variable (whose contrast with the stock is precisely the absence of a Relationship from its own prior state), the cloud or source/sink (the reified other of the boundary Distinction, represented by the familiar SD cloud notation), the conveyor and the aging chain (order within and between stocks), conserved flows (part-accounting across wholes), nonlinear Relationships (Forrester's own nomination [19]), and the mental model—whose canonical definition, a representation "whose structure is analogous to the perceived structure of that system" [18], asserts in the field's own words the mind-nature structural correspondence that DSRP claims.

The full instrument—rule, rulings, proposed decompositions, and exclusion log—is available upon request as supplementary material. The containment claim therefore stood exposed on twenty-four named fronts: a single one of them yielding a residual X would have falsified it. Section 16 records the result: none did.

16. Sixty Constructs, Zero Residuals

Sections 4 through 13 decomposed twenty-one of the sixty constructs extracted under the rule of Section 15; twelve more were identified there as composites of already-decomposed elements and three as purely informational. This section completes the test by decomposing the remaining twenty-four. Each entry names the construct, with aliases merged from the glossary [18], states its necessary organizational content, performs the decomposition under the decision rule of Section 2, and closes with the same verdict language used in Table 1. The entries are grouped as structural machinery (16.1–16.12), patterns across models (16.13–16.15), behavior (16.16–16.18), and epistemic constructs (16.19–16.24). One entry—16.9—deserves special notice because nonlinear Relationships present one of the strongest apparent candidates for a structural residual.

16.1 Auxiliary (converter) variable. An auxiliary is a distinguished identity carrying a value that is recomputed at every instant from other current values via its incoming connectors; it is "capable of changing its value instantaneously" [18]. Its structural interest lies in the contrast with the stock. A stock carries accumulated state: its next value depends on its prior value together with its inflows and outflows. An auxiliary has no corresponding accumulation of its own prior state; its current value is determined by its current inputs. The stock/auxiliary distinction is therefore organizational as well as informational: the stock is a persistent quantified System whose state is changed through flow Relationships, whereas the auxiliary is a distinguished value-bearing object computed through its incoming Relationships. Residual organizational primitive: none identified.

16.2 Connector (information link). A connector is a Relationship that carries information rather than material, and it obeys a constraint: it may terminate on auxiliaries and flows, "but never stocks" [18]. Two decomposable features are present. The material/information typing of Relationships is a Distinction drawn on Relationship objects themselves—the Relationship-as-object move of Section 5. The targeting constraint specifies which Relationships may exist between which parts of the model-System. Residual organizational primitive: none identified.

16.3 Cloud (source, sink). The cloud is one of the more conceptually sophisticated pieces of SD notation. It explicitly acknowledges that the modeled System has an outside: flows may originate beyond the model boundary or disappear beyond it, and the model does not pretend to represent everything. In that sense the cloud encodes a form of epistemic humility. Structurally, the cloud externalizes the other of the model-level Distinction:

$$D_{model} = \{i_{model}, \mathbf{r}_{io}, o_{model}\}$$

where i_{model} is the modeled whole and o_{model} is what lies outside it. But the cloud applies this insight principally at the scale of the model-System. DSRP-484 generalizes the same structure to every distinguished object. For every identity i_k there is a coimplied other o_k and therefore:

$$D_k = \{i_k, \mathbf{r}_{io}, o_k\}, \quad k = 1, \dots, n$$

Thus a model containing n distinguished identities contains n corresponding identity-other structures:

$$n_i \rightarrow n_o$$

The SD cloud therefore represents one special case of a universal boundary condition. SD makes the outside of the model explicit; DSRP predicts an other for every distinguished thing within it. This distinction matters because each local other contributes to the contextual identity of its corresponding object. "Inventory," "population," "demand," "workforce," and every other modeled construct are defined not merely by their names but by what they exclude, what they relate to and do not relate to, which wholes they belong to, and which structures frame or are framed by them. The advanced Is/Is Not operation reconstructs those contextual identities by assembling and counting such structural statements. The cloud is therefore not wrong but under-generalized. It makes model-level externality visible, while DSRP-484 predicts recursively distributed identity-other structure at every scale. Two consequences follow. First, the cloud is the not-System. Every System coimplies a Distinction, $S_{pw} \rightarrow D_i \mid D_o$, and what the cloud draws is the D_o : the other of the modeled whole, rendered as notation. A System cannot be drawn without its Distinction, and system dynamics has been drawing both since Forrester. Second, the excluded side is not empty. What the cloud marks is the model's context — the organization outside the boundary on which the model's information silently depends [11]. Residual organizational primitive: none identified.

16.4 Conserved flow. A conserved flow moves a quantity between stocks "so that the total amount of material in the related part of the system is unchanged" [18]. Conservation is a part-accounting constraint across wholes: quantity leaving one stock is accounted for as quantity arriving in another, while the transfer occurs through a Relationship between Systems. Where individual parts are explicitly represented, their identities may also persist through the transfer; where the stock is continuously measured, conservation operates on the quantified whole without requiring individual parts to be enumerated. A non-conserved flow, by contrast, crosses the model boundary to or from a cloud, where the accounting deliberately stops. Conservation therefore requires no new organization beyond distinguished content, wholes, transfer Relationships, and the accounting frame under which conservation is asserted. Residual organizational primitive: none identified.

16.5 Conveyor (pipeline delay). A conveyor is a stock whose contents do not mix: a quantity entering at time t exits after the specified transit interval [18]. Exit order equals entry order. Structurally, a conveyor is an ordered whole: its contents occupy ordered positions or states connected through directed Relationships, and that internal ordering is preserved through the outflow. Transit time is information carried under the temporal Perspective developed in Section 9. The contrast with an ordinary perfectly mixed stock is therefore the presence or absence of ordering structure among the contents of the whole, generated by composing directed Relationships into ordered chains. The conveyor thereby tests the calculus's ordering machinery within a stock, not merely between events. Residual organizational primitive: none identified.

16.6 Aging chain. An aging chain is "a sequence of stocks connected by conserved flows" whose contents "move in one direction" [18]. Each stage is a bounded whole organized through D and S ; the

stages are linked by conserved directed Relationships R ; the chain as a whole is a higher-order System whose parts are those stocks and Relationships; and one-directionality follows from the orientation of the constituent Relationships. Nothing in the chain exceeds recursive composition of structures already decomposed. Residual organizational primitive: none identified.

16.7 Co-flow. A co-flow is "a parallel stock-and-flow structure that mimics a primary stock-and-flow structure," tracking "an attribute or characteristic of the contents of the primary structure" [18]. Three pieces are present: the primary $S(R)$ structure; a second $S(R)$ structure carrying attribute information about the primary structure's contents; and the correspondence between them—a Relationship under which changes in the primary structure are associated with corresponding changes in the co-flow structure. Structures related to structures are recursion, not novelty. Residual organizational primitive: none identified.

16.8 Closed system. A closed system "functions without the influence of exogenous variables" [18]. Structurally, it is a bounded System whose relevant generating Relationships are enclosed within the whole and do not cross the specified model boundary. This is endogeneity stated as a structural condition: $D+S$ with a closure constraint on R , asserted under P . Ford notes that "a completely closed system does not exist in reality" [18]; the construct is therefore a limiting case, and limiting cases inherit the decomposition of what they limit. Residual organizational primitive: none identified.

16.9 Nonlinear Relationship. A nonlinear Relationship is one in which "the change in the impacted variable is not directly proportional to the change in the impacting variable" [18]. Forrester himself identified "high-order nonlinear feedback structures" as central to the field's future [19]. Yet a generic Relationship never requires proportionality: a acts and r reacts, but nothing in R_{ar} requires the reaction to remain a constant multiple of the action. Linearity is therefore a particular informational constraint on a Relationship; nonlinearity permits the quantitative mapping carried by that Relationship to vary with state. The mathematics may become substantially more difficult—superposition can fail and closed-form solutions may disappear—but mathematical difficulty is not evidence of a new organizational primitive. State-dependent relational information can also contribute to higher-order organizational changes such as shifts in loop dominance, providing one route for the $\Delta I \rightarrow \Delta O$ coupling described in Section 11. The burden therefore reverses. A generic Relationship never promised proportionality, so a critic seeking a residual at this construct must argue that proportionality, of all things, is organizational — a burden we are content to leave with them.. Residual organizational primitive: none identified.

16.10 Graphical function (table function, look-up function). A graphical function specifies a Relationship extensionally—as plotted point-pairs or a table—rather than intensionally through a closed-form formula [18]. A table, graph, and equation are themselves differently organized representational objects, as Section 11 makes explicit. But at the level of the modeled SD Relationship, changing among these representational forms need not change which relata are related or the organizational role of that Relationship. The mapping is represented through different mathematical or graphical organizations without introducing a new primitive elemental into the organization of the modeled system. Residual organizational primitive: none identified.

16.11 System order (high-order system). The order of an SD model is the count of its stocks—oscillation, for instance, requires "a second-order or a higher-order system" [18], and Forrester's phrase makes high order constitutive of the subject matter [19]. The count presupposes that each stock is distinguished and treated as a state-carrying sub-System. The numerical count itself is information, while the distinctions among stocks, their organization within the larger model-System, and the Perspective under which that level of aggregation is selected are organizational. Change the aggregation level—for example, by splitting one workforce stock into three departmental stocks—and the same target phenomenon can be represented at a different model order. Stated in the calculus, order is a Perspective taken on the model: the point is ORDER, $P_p(\text{ORDER})$, and the view is the number of sub-Systems that carry state. A model's order is therefore not simply an absolute property of the target system; it is what the model shows from the vantage that asks how many state-carrying wholes there are. Residual organizational primitive: none identified.

16.12 Policy (decision function). A policy is "a decision rule or structure that uses information streams to generate decisions" [18]. Structurally it is a reified Relationship: a named, inspectable, comparable transformation from informational inputs to actions upon flows. Its algorithm is itself organized and can be decomposed into ordered Relationships among distinguished states and operations; its inputs are information I carried by connectors; its outputs act on quantified Relationships; and its reification—the fact that a policy can itself be distinguished, compared, and redesigned—is the Distinction-on-Relationship move of Section 5. That policies are objects is what makes policy comparison and redesign possible. Residual organizational primitive: none identified.

16.13 Generic structure (transferable structure). A generic structure "can be applied across different settings due to having the same fundamental underlying components and relationships" [18]. Read through O-Theory, this construct exploits the distinction between relatively invariant organization and variable information. Different domains can instantiate different informational content within sufficiently corresponding organizational structures. In compact form, the generic structure is a case of different I instantiated within sufficiently corresponding O . Identifying such a structure requires a Distinction among structures and a Perspective that abstracts away domain-specific content. The construct's usefulness therefore depends directly on the separability of information and organization developed in Section 2. Residual organizational primitive: none identified.

16.14 System archetype. An archetype is "an integrated feedback structure, the resulting behavior mode or modes, and a story of how the structure can create the behavior modes" [18]. It packages three already-decomposed things: a generic $S(R)$ configuration, its behavior-over-time signature, and a narrative or explanatory frame supplied under a Perspective. Residual organizational primitive: none identified.

16.15 High-leverage point. A high-leverage point is a part of the system "where small changes can have a very large impact on system behavior" [18]. It is a locus—a Distinction drawn within the model-whole—whose salience is relational and comparative: how much does changing this location alter some behavior of interest? That comparison is evaluated under a Perspective. Where the intervention changes a parameter or value, leverage concerns sensitivity to ΔI ; where it changes structure, it concerns sensitivity to ΔO . The coupling developed in Section 11 permits both. Residual organizational primitive: none identified.

16.16 Behavior mode. A behavior mode is "a shape or pattern over time of the values of a system variable" [18]. The values are information I indexed through time. TIME supplies the temporal Perspective P_t under which the trajectory is considered, while the ordering of states can be represented through the directed relational structure discussed in Section 9. Identifying a particular shape—GROWTH, DECLINE, OSCILLATION, OVERSHOOT, and so forth—is itself a Perspective taken on that trajectory: the point is a SHAPE, $P_p(\text{S-SHAPED})$ or $P_p(\text{OSCILLATION})$, and the view is the pattern the time series presents. A taxonomy of behavior modes is simultaneously a System whose parts are these distinguished pattern-views, each of which sees its own shape in the data and ignores the others. Naming a mode does not add a new primitive to the structure that produced the trajectory; it frames the trajectory in a particular way. Residual organizational primitive: none identified.

16.17 Equilibrium (equilibrium behavior, equilibrium conditions, stasis, steady state). Equilibrium is a condition in which "the inflows and outflows of each stock balance each other, and the sizes of the stocks do not change" [18]. It is principally an informational condition on standing organization: the quantities carried by incoming and outgoing Relationships balance so that the quantity carried by the stock-System does not change. In dynamic equilibrium the flows may continue and the contents may change while the whole's measured quantity remains constant—Section 7's persistence argument in miniature. Residual organizational primitive: none identified.

16.18 Stability and instability (stable/unstable behavior, stable/unstable equilibrium). Stability concerns how a System responds after disturbance: whether it tends to return toward equilibrium conditions or move further away [18]. A disturbance may involve informational change, ΔI , organizational change, ΔO , or both. The subsequent trajectory is generated through the standing organization and information of the model, while its classification as stable or unstable is made relative to specified limits or criteria and therefore under a Perspective. Stability is thus a derived property of information and organization under a frame, not an additional primitive elemental. Residual organizational primitive: none identified.

16.19 Mental model. The glossary defines a mental model as an enduring internal representation "whose structure is analogous to the perceived structure of that system" [18]. Note what this definition asserts: minds hold organized structure, and that mind-side organization can correspond to perceived system-side organization. In the notation of Section 2, a mental model is an organized cognitive object:

$$\mathbb{M} = I_{\mathbb{M}} \cdot O_{\mathbb{M}}$$

Its organization includes Distinctions, part-whole Systems, Relationships, and Perspectives held by a cognitive agent about a target. The construct is therefore not merely containable within DSRP-484; its canonical SD definition independently invokes the kind of mind-reality structural correspondence at issue in the DSRP research program [8,10]. Residual organizational primitive: none identified.

16.20 Dynamic hypothesis. A dynamic hypothesis is "a structure that the modeler advances to explain a dynamic behavior of interest" [18]. It is a claim, made under a Perspective, that a proposed organization generates or explains an observed informational trajectory. The hypothesis therefore establishes a

Relationship between a candidate organization O and a target behavior, framed by the modeler's explanatory purpose P . Residual organizational primitive: none identified.

16.21 Reference mode. A reference mode is a behavior-over-time pattern "used in problem articulation" and "in model validation" [18]. It is a distinguished time-indexed informational trajectory selected under a Perspective and elevated into a criterion against which the problem is framed or the model compared. Its informational values I_i are considered under a temporal Perspective P_i , while selection of that particular trajectory as the relevant reference is another Perspective operation. Residual organizational primitive: none identified.

16.22 Model credibility (validity). Credibility is "how well a model represents a given problem; a model's suitability for a particular purpose" [18]. The testing workflow itself is excluded under the rule of Section 15 as modeling practice; the relation that workflow estimates is included. Credibility decomposes as a comparative Relationship between the model's information-and-organization and the target's information-and-organization, judged relative to a purpose. In O-Theory terms, the comparison concerns organized phenomena of the general form $X=I \cdot O$. The phrase "suitability for a particular purpose" makes the Perspective explicit: representation quality is evaluated from a specified point of concern rather than as purpose-free correspondence. Residual organizational primitive: none identified.

16.23 Systems thinking. The glossary's account of systems thinking has practitioners "see things on three levels: events, patterns of behavior and system structure" [18]. These levels align closely with the distinction developed in this paper. Events foreground informational states; patterns of behavior organize those informational states into trajectories through time; and system structure foregrounds the organization that generates those trajectories. The levels are not mutually exclusive—events and patterns are themselves organized—but they represent progressively deeper attention to the organization underlying observed information. As a practice, systems thinking also involves repeated Perspective-taking: seeking loops rather than isolated lines, processes rather than snapshots, and endogenous structures rather than merely external events [18]. Residual organizational primitive: none identified.

16.24 Dynamic thinking, stock-and-flow thinking, operational thinking. Richardson names these among the skills that expanded Forrester's founding threads [15]. Each is principally a trained Perspective directed toward structures already decomposed above. Dynamic thinking frames information in terms of change through time; stock-and-flow thinking frames persisting wholes and the Relationships through which their quantities change; operational thinking asks what mechanism lies inside an apparent Relationship—the opening of R into a System composed of multiple Relationships:

$$S(R_1, \dots, R_n)$$

These thinking skills therefore apply particular Perspectives to already-decomposed objects rather than introducing new organizational primitives. Residual organizational primitive: none identified.

16.25 Result. Twenty-four constructs were attempted; twenty-four decomposed; no residual X was identified. Combined with Sections 4 through 13, the tally over the full pre-registered candidate set stands

at sixty of sixty, with the seventeen logged exclusions available for audit in the supplementary material. A notable pattern in the rulings is that several constructs SD often describes as properties of models—including model order, behavior mode, credibility, and even the framing implicit in what a model places inside or outside its boundary—contain substantial Perspective structure. The field's vocabulary therefore contains more *P*-structure than its ordinary self-description makes explicit, consistent with the DSRP-484 claim that Perspective is recursively enclosed within organization [7]. That structure is not merely descriptive. As Section 9 shows, a Perspective whose point is an ordering point coimplies an ordered view, so Perspective can generate organization rather than only frame what is already organized. The Perspective structure recovered across these rulings is therefore productive, not decorative — which is part of why it goes unnoticed in a vocabulary that treats model order, behavior mode, and credibility as objective properties. The claim of Section 17 survives this test not because the test was incapable of failure, but because construct by construct—including nonlinear Relationships, one of the strongest apparent candidates for a residual—nothing organizational remained unexplained.

17. Summary of Findings

This paper asked a simple question: are the building blocks of system dynamics themselves made of something more basic?

To answer it, the test had to be able to fail. Before decomposing the constructs, we fixed the rule for what would count as an SD construct and drew the candidate set from documents the field wrote about itself—its official glossary, Richardson's account of its foundations, and Forrester's statement of its subject matter. We added nothing to the list ourselves. Seventeen entries were excluded under pre-declared rules, with every exclusion logged. The remaining sixty constructs were then taken apart to see whether any required a kind of organization that DSRP-484 could not account for.

One such remainder would have ended the argument. None appeared. Sixty constructs were tested; all sixty decomposed without requiring an additional organizational primitive. Twenty-one were decomposed in Sections 4 through 13 and twenty-four in Section 16; twelve were combinations of structures already decomposed, and three concerned only the information or values carried by a model rather than additional organization.

What does that mean in ordinary terms?

A variable begins with a Distinction: something has been identified as this rather than everything it is not. A stock is a part-whole System carrying a quantity; the whole can persist while its parts and quantity change, which is why the bathtub remains the bathtub while water enters and leaves. A flow is a quantified Relationship through which the state of a stock changes. A feedback loop is a System composed of Relationships arranged so that a causal path closes back upon itself. A delay is either temporal information carried by a Relationship or, when opened at finer resolution, a process-System composed of additional Relationships. A boundary distinguishes inside from outside, organizes what is inside as a whole, and depends on the Perspective from which that boundary is drawn. The familiar SD cloud makes the outside of that model-level Distinction visible.

None of these constructs required a new kind of organization.

Three additional findings emerged from the decomposition.

First, several constructs that SD often treats as straightforward properties of a model contain more Perspective structure than their usual descriptions acknowledge. Model order depends on how the model has been aggregated. A behavior mode depends on which pattern is being recognized in a trajectory. Credibility is explicitly judged relative to a purpose. Even a model boundary depends on what the modeler has chosen to treat as inside, outside, and relevant. Perspective is therefore already present throughout SD, even where it is not named.

Second, the construct that looked most likely to produce a residual did not. Forrester identified "high-order nonlinear feedback structures" as central to system dynamics, making nonlinearity an obvious place to look for something organizationally irreducible. But a Relationship does not require proportionality in the first place. Linearity is the special case in which an additional proportional constraint has been imposed. Removing that constraint produces nonlinearity without requiring a new organizational primitive.

Third, the field's own definition of a mental model closely echoes the premise being tested here. The SD glossary defines a mental model as a representation "whose structure is analogous to the perceived structure of that system." In other words, the field itself describes a correspondence between organized structure in mind and organized structure in the world—the same general mind-reality correspondence proposed by DSRP.

It is important to be precise about how strong the main result is. Several weaker claims were possible.

The weakest would be compatibility: DSRP and SD can be used together without contradiction.

$$DSRP \leftrightarrow SD$$

A stronger claim would be representability: the structures of SD can be described using DSRP.

$$SD \rightarrow DSRP$$

The result here is stronger than either. Every organizational structure distinctive to system dynamics in the sixty-construct candidate set could be built from DSRP-484, while DSRP-484 also describes organizational structures that are not part of system dynamics. At the organizational level tested here, SD therefore occupies only part of the larger space described by DSRP-484. This is what is meant by strict structural containment. The reverse does not hold. SD provides no corresponding general calculus of Distinctions, Systems, Relationships, and Perspectives from which the full DSRP organizational space could be generated.

$$O_{SD} \subsetneq Cl(O_{DSRP-484})$$

$$\exists c \in C_{SD} \text{ such that } O_c \notin Cl(O_{DSRP-484})$$

By Occam's razor, that matters. If DSRP-484 accounts for all of SD's organizational structures with fewer and more general primitives, while also accounting for structures beyond SD, then DSRP-484 is the more parsimonious organizational theory. In other words, DSRP-484 can do everything SD can do. SD cannot do everything DSRP-484 can.

That turns the original question around. The question was never whether DSRP can "also do" feedback, stocks, flows, and delays. The question is whether feedback, stocks, flows, and delays were DSRP structures all along.

The result remains falsifiable. One necessary SD construct whose organization cannot be built from Distinctions, Systems, Relationships, and Perspectives would break the containment claim.

The full instrument—the extraction rule, candidate set, decomposition of every construct, rulings, and exclusion log—is available so that anyone who wants to challenge the result can begin with the same test.

18. Discussion: What Is System Dynamics For?

18.1 Taking stock

A result like this can easily be read as a scoreboard: DSRP wins, system dynamics loses. That would miss the point.

The purpose is not to rank methods. It is to understand the systems people live and work in and to put better thinking tools in more hands. More than eight billion people make decisions every day inside systems they did not design and cannot see whole. The practical question is therefore not which method wins, but which tools help people see more clearly and act more effectively.

Systems thinking, like every mature field, has tribes. Tribes help methods develop, spread, and survive. They also create identity. People invest years learning a method, teaching it, publishing with it, building communities around it, and becoming known for using it well. A claim about the foundations of that method can therefore be heard as a claim about the value of the people who practice it. If someone has spent thirty years building system dynamics models, hearing that stocks, flows, and feedback loops are composites may sound like being told that those thirty years were spent learning the wrong thing.

That is not the finding.

Discovering that a useful construct is composite does not make it less useful. Chemistry did not become less important when molecules were shown to be composed of atoms. The explanatory level changed; the value of the higher-order structure did not. The same is true here.

Nothing in the preceding analysis shows that system dynamics is wrong, unnecessary, or should be used less. It shows where its constructs sit organizationally. Stocks, flows, feedback loops, delays, and the rest remain useful precisely because higher-order composites can be enormously useful. The question is what those composites are good for, what they cost, and what happens when they are used outside the class of problems for which they were designed.

18.2 What system dynamics buys you—and what it costs

System dynamics can do something extraordinarily useful: it can produce a model that a computer can run.

That is not a trivial achievement. Calculus does not tell a modeler that workforce should be treated as an accumulation and hiring as a rate. DSRP does not tell the modeler that either. What system dynamics supplies is a modeling convention: treat this whole as a stock, treat these Relationships as flows, connect them through feedback, quantify the rates, and the resulting structure can be translated into equations and simulated.

That convention is powerful because it restricts the problem.

A quantitative SD model cannot represent a phenomenon in just any form. The phenomenon must be cast into stocks, flows, causal Relationships, feedback structures, and related constructs that can be quantified and executed. That restriction is part of what gives system dynamics its rigor. The model must compile. Units must balance. Stocks must integrate their rates. The behavior generated by the model can then be compared with observed or expected behavior.

But every restriction creates a blind spot.

A stock, in the organizational sense developed in Section 7, is a whole whose quantity changes as things enter or leave it. Population is people. Inventory is items. Backlog is orders. Cash is units of currency. These are natural fits.

Other cases are less natural. When morale is modeled as a stock, for example, the modeler has chosen to treat morale as though some measurable quantity of it accumulates and drains through rates. That may be a useful idealization. But the formalism itself cannot tell us whether it is a good one. Once morale has been declared a stock, SD knows what to do with it.

The danger therefore begins before simulation.

The formalism does not first ask, "What organization does this phenomenon actually have?" It asks how the phenomenon can be expressed using the organizational forms SD provides. DSRP begins one level earlier. It does not pre-select stocks, flows, feedback loops, or delays; it asks what Distinctions, Systems, Relationships, and Perspectives the phenomenon actually requires. That is the practical consequence of treating DSRP-484 as a candidate universal organizational calculus rather than as another domain-specific modeling language.

That distinction matters because systems do not arrive with one privileged Perspective attached to them. The same system can be organized and understood from many Perspectives. Importantly, those perspectives are inside the system, not outside of it. System dynamics selects an external System of Perspectives ("S of P move") in advance: look for ACCUMULATION, FLOW, FEEDBACK, DELAY, and CHANGE OVER TIME.

When that Perspective fits the problem, SD is exceptionally useful. When it does not, the method can make the problem fit the Perspective. That is the cost of the convention.

18.3 The elk

Consider an elk herd.

From a POPULATION Perspective, the herd is almost a perfect system dynamics problem. Births flow in. Deaths flow out. The herd accumulates. Food availability affects reproduction and mortality. Those changes feed back into population size.

A good SD model can tell us things about those dynamics that conceptual analysis alone cannot.

Now consider the same elk herd from another Perspective.

Does an elk distinguish one grass as FOOD and another as NOT-FOOD? Does it PREFER one species of grass over another? Does that PREFERENCE change by SEASON, AGE, LOCATION, NUTRITIONAL STATE, or WHAT OTHER FORAGE IS AVAILABLE?

The subject has not changed. The elk are the same elk. What changed is the Perspective. "Food from the elk's point of view" sets the Perspective. "Is this grass food?" requires a Distinction. "What grasses are included in elk food?" requires a System. "Do elk have different relationships with different grasses at different times?" requires Relationships.

Different grasses may form Systems of alternatives. Preference establishes Relationships among them. Those structures may later be quantified, and some may ultimately be represented in an SD model. But SD does not tell the analyst which Distinctions should be made, which Perspectives matter, or which nested structures and relationships should be examined before quantification begins.

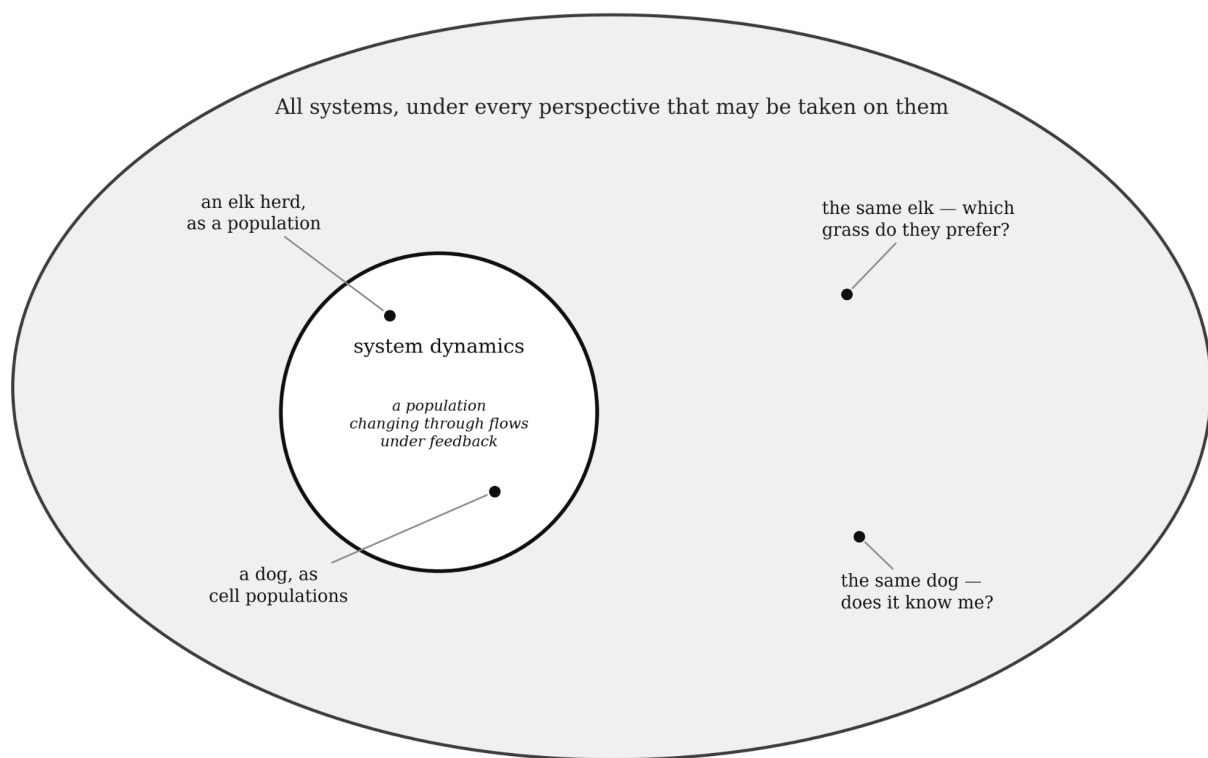


Figure 2. Venn diagram of SD Utility Compared to All Systems. The outer region represents the larger space of systems together with the Perspectives that may be taken on them. The inner region represents the subset naturally expressible through the stock-flow-feedback structures of system dynamics. The elk herd appears twice because the system is the same while the Perspective changes. The regions are schematic and do not imply a measured proportion.

That is the gap.

System dynamics is excellent at quantifying distinctions and structures once they have been specified. It is not a general calculus for discovering and interrogating the organization from which those modeling choices arise.

The restriction is therefore not primarily about subject matter. Elk are not "inside SD" while some other topic is outside it. The same elk can fall inside or outside the natural reach of SD depending on the question being asked— the Perspective taken from them.

The relevant unit is therefore closer to: SYSTEMS x PERSPECTIVES. A POPULATION Perspective on the elk fits SD naturally. A FORAGE-PERCEPTION Perspective may not.

This is why the limitation of SD can be difficult to see from inside the method. Once the POPULATION Perspective has been adopted, the world presents exactly the kinds of things system dynamics is designed to model. This has a name: *confirmation bias*.

This tells us where SD's greatest strength lies. But it also shows us where its weakness becomes the user's bias. In Figure 3 we see the comparative coverage that SD provides in understanding systems.

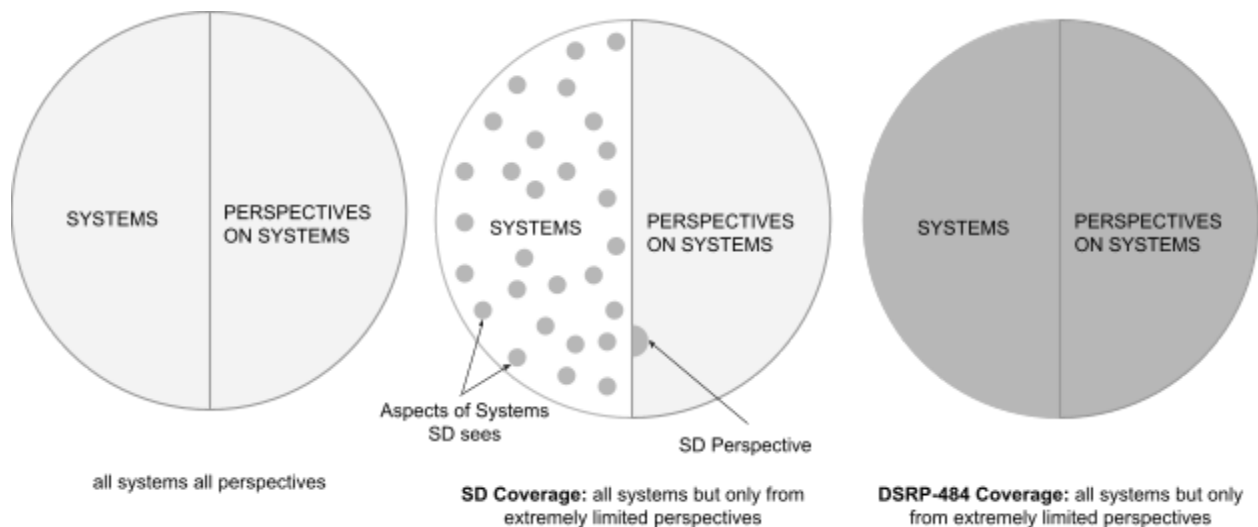


Figure 3. Many Systems from Many Perspectives. A split Venn diagram of all of the Systems in the world and all of the Perspectives from which one could view those systems. Understanding Systems requires us to look at them from the most salient perspectives, which may or may not be the perspective "baked in" to SD. SD can simulate Systems, but the cost of that simulation is a serious reduction in the number of Perspectives and Systems that can be simulated reliably. SD covers all systems but only from a very small and very limited perspective. DSRP-484 covers all systems and all perspectives.

The important question is what was not seen because another Perspective was never taken.

Figures 2 and 3 make a ratio visible. Let ASP be the set of all System–Perspective pairs that systems thinking can address, and SDA the subset appropriate to system dynamics — those expressible in stock-flow-feedback form. The opportunity cost of system dynamics is then:

$$\frac{|\text{SD-Appropriate System–Perspective Pairs (SDA)}|}{|\text{All System–Perspective Pairs (ASP)}|} = \frac{|\text{SDA}|}{|\text{ASP}|} \ll 1$$

$$\text{SDA} \subsetneq \text{ASP}$$

The containment is strict and the disparity is large. This is the same shape as the paper's structural result, arriving now at the level of practice rather than theory: the space SD generates lies inside the space, and does not fill it.

The cost is therefore not SDA. It is $\text{ASP} \setminus \text{SDA}$ — the pairs that a commitment to the formalism puts out of view. That cost is only incurred when system dynamics is treated not as one method among several but as systems thinking itself. Used inside its range, SD pays nothing. Used as the general instrument, it pays the difference. This applies to *quantitative* SD. *Qualitative* SD — the bulk of SD activity in practice — poses a harder problem (§18.4).

18.4 The two system dynamics

There is a second reason this matters, and it concerns how system dynamics is actually *practiced*.

Quantitative system dynamics earns its credibility from a discipline. The model must compile. Units must balance. Stocks must integrate their rates. That discipline is inseparable from the restriction just described: only what can be represented as a population changing through feedback can be modeled at all. Real rigor, bounded domain.

Qualitative system dynamics — causal loop diagramming without simulation — keeps the *reputation* but lacks the *discipline*. It travels to organizational culture, public policy, conflict, education, and health systems, domains where in many cases nothing would have compiled, carrying authority earned where things did compile. Those are very different situations and different contexts.

The field knows this and has argued about it for a quarter century. Coyle argued that quantification can mislead badly when soft variables compound, and that diagrammatic work is sometimes the honest stopping point [20]. Homer and Oliva replied that simulation nearly always adds value and that maps are unreliable instruments for inferring behavior [21]. The dispute has not settled; System Dynamics Review devoted a seventeen-paper special issue to qualitative aspects of the method in 2024 [22]. This is not an outside criticism. It is the field's own longest-running internal question.

We would add one observation to it. *The quantitative restriction announces itself; the qualitative one does not*. If a phenomenon is not a population changing through flows, the model will not run, and the machinery says so. Nothing whatever stops anyone from drawing a causal loop diagram of anything at all. The precise feature that made system dynamics rigorous is the feature that disappears, without an error message, the moment one drops to maps.

And there is something in Homer and Oliva's argument that reaches past the dispute they were having. Defending simulation, they observe that people "do a poor job of mental simulation even when they have complete knowledge of system structure" [21]. Complete knowledge of the structure, and still not the behavior. That is the distinction between organization and information, drawn by the method's most orthodox defenders in the course of defending it. The map carries the organization. The simulation supplies the trajectory that organization generates once information is loaded into it.

This also helps explain why causal loop diagrams travel so easily across domains.

When no simulation is intended, much of what the practitioner is actually doing is organizational thinking: distinguishing things and drawing Relationships among them.

Organization is universal, so the diagrams travel.

But the qualitative SD tool is structurally thin. It makes feedback highly explicit while leaving much of Distinction, System, Perspective and the deeper features of Relationships implicit. The analyst draws what affects what, but may never systematically ask what each thing is and is not, zoom out to larger wholes, zoom into the parts, zoom in to the relationships to see what's really going on beyond flows and rates, or consider the dynamic effects of shifting perspective.

The universality of qualitative system dynamics is therefore not an embarrassment to the containment result. It is evidence for it. The part that generalizes across domains is precisely the part that is not uniquely system dynamics: the underlying organizational work. DSRP represents that work with greater fidelity.

18.5 What follows for practice

The practical lesson is not "use DSRP instead of system dynamics." It is more useful than that: determine whether the system, under the Perspective that matters, is actually an SD problem before using SD.

When it is, system dynamics is extraordinarily powerful. When it is not, system dynamics is not a neutral modeling language. Without DSRP, SD modeling becomes SD bias.

That distinction should change the order in which modeling is done.

Before drawing a stock, flow, or causal loop, the analyst should first ask what organization the phenomenon actually contains. What has been distinguished, and from what? What are the relevant wholes and parts? What Relationships exist or may be missing? From whose or what Perspective is the phenomenon being framed? What alternative boundaries, decompositions, Relationships, and Perspectives would produce a different model? What structures lie inside a relation or a delay? And what organizational assumptions were frozen before the equations were written?

These are upstream questions. SD does not answer them automatically because its vocabulary already presupposes a particular way of organizing the phenomenon.

DSRP-484 is useful at precisely this point. It provides a way to examine organization before committing to a domain-specific modeling form. The purpose is not to force DSRP into the final model. It is to prevent the final model from being determined prematurely by the tool chosen to build it.

The first practical decision is therefore straightforward.

If the phenomenon of interest can defensibly be represented as one or more accumulations whose quantities change through flows, with feedback and delay materially affecting behavior over time, then use system dynamics. Build the stocks and flows. Estimate the rates. Specify the feedback structure. Check the units. Run the simulation. Test the behavior. This is what SD was built to do, and within that problem class it is exceptionally powerful.

Epidemiology, inventory and supply chains, demography, hydrology, ecological populations, workforce dynamics, and many project-resource problems naturally contain this structure.

But if the relevant phenomenon does not naturally have that organization, stop.

Do not turn something into a stock merely because SD requires a stock. Do not invent a flow because the formalism expects one. Do not assume that every important Relationship belongs in a feedback loop. Do not treat the absence of an SD representation as evidence that the phenomenon is unimportant. And do not confuse a model that runs with a model of the right thing.

The same rule clarifies the role of causal loop diagrams. If causal loop diagrams are being used for elicitation, exploration, or communication rather than prediction, then the task is fundamentally organizational. The analyst is deciding what things exist in the model, how they are distinguished, how they relate, what wholes they form, where boundaries fall, and which Perspectives matter.

DSRP-484 provides a more general instrument for that task because it makes all four forms of organization explicit rather than privileging Relationships and relational closure.

If causal loop diagrams are instead being used to infer dynamic behavior, the standard SD objection applies: people are poor at mentally simulating complex feedback systems, so the model must eventually be formalized and simulated. But even then the analyst still faces the prior question: was the right organization put into the model before the equations were written?

There are therefore two distinct ways an SD analysis can fail.

The first is a model that runs but models the wrong organization.

The units balance. The equations are valid. The simulation behaves plausibly. The model may even reproduce the reference mode. But a consequential Distinction was never made, an important Perspective

was omitted, the boundary was drawn badly, a Relationship was represented too crudely, or several structures that should have remained separate were aggregated into one variable or stock.

Nothing inside the solver will identify those mistakes because they occurred before the solver became involved. A mathematically correct simulation of the wrong system can be more dangerous than an obviously broken model because numerical precision gives the result an appearance of authority.

The second failure is more basic: the phenomenon should never have been represented as an SD problem at all.

It was not naturally an accumulation changing through flows under feedback, but it was forced into that form because SD was the method selected in advance. By the time the model runs, the consequential bias has already been built into its organization.

Both failures arise from the same reversal of method and inquiry: choosing the modeling language first and then organizing the world so that the language can describe it.

Caddell and Nilchiani [23] demonstrate the opposite sequence. In developing their model of interpersonal trust, they did not begin by immediately translating trust into stocks, flows, and feedback loops. They first used DSRP to map the literature and identify relevant Distinctions, Systems, Relationships, and dependencies. Only then did they translate the resulting organization into a system dynamics model for mathematical analysis and simulation.

That sequence matters. DSRP did not replace SD. It helped determine what the SD model should be a model of.

But DSRP need not disappear once SD has been selected. The modeler must realize that it is the DSRP ontology that is driving the model.

DSRP can also improve the SD model itself. Relationships are a particularly important example. SD commonly represents a causal Relationship as a link between variables. DSRP treats the Relationship as an organized object in its own right, with action and reaction structure that can itself be distinguished, decomposed, related, nested, and viewed from different Perspectives. That additional resolution can clarify what is actually acting on what, distinguish the relata from the Relationship connecting them, reveal missing intermediate structure, and expose when a seemingly simple causal link should be opened into a System of Relationships.

The same applies to the other Patterns.

Distinctions can sharpen variable definitions by making explicit not only what a variable is, but what it is not, where its boundary lies, and what has been excluded from it.

Systems can improve aggregation and disaggregation by forcing the analyst to examine which parts constitute which wholes, what has been collapsed into a stock, what should remain separate, and how structures are nested across scale.

Relationships can improve causal specification by examining the organization inside each link rather than treating an arrow as self-explanatory.

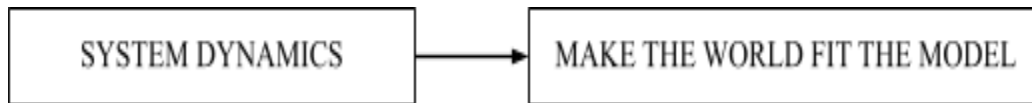
Perspectives can expose whose or what frame determines relevance, boundary placement, variable selection, interpretation, and the behavior being treated as important.

DSRP therefore has two distinct roles in relation to system dynamics.

First, it operates upstream of SD by helping determine whether the phenomenon is appropriately modeled using system dynamics at all.

Second, when SD is appropriate, DSRP operates within the modeling process by helping make the SD model more structurally precise.

The workflow is therefore not:



It is:



Use DSRP first to determine what organization the phenomenon actually contains. Then ask whether the resulting system-and-Perspective is genuinely suited to stock-flow-feedback modeling. If it is, use SD aggressively—and continue using DSRP to sharpen the Distinctions, Systems, Relationships, and Perspectives from which the model is constructed.

If system dynamics is appropriate:

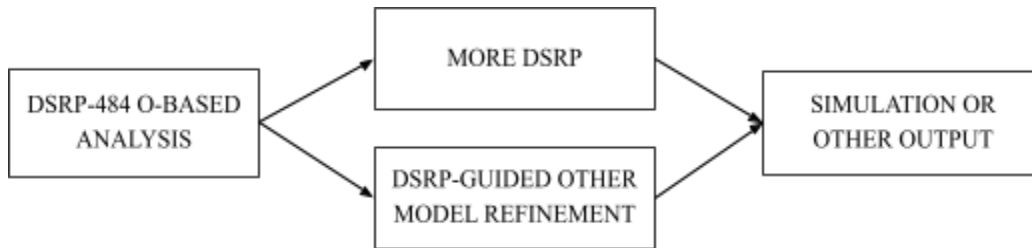


The recommendation is therefore not to use less system dynamics, but to use it when it is the best tool for the job — and the job is determined by the system and the Perspective. When the system and Perspective are right for SD, SD is powerful. When they are not, SD is not merely less useful. Its stock-flow-feedback frame becomes a bias: it makes some organization easy to see, other

organization difficult to see, and can quietly transform the phenomenon into the kind of system the method knows how to model.

The rule is therefore simple. Do not begin with system dynamics and ask how the world can be represented in it. Begin with the organization of the world using DSRP, determine whether system dynamics fits it, and, when it does, use DSRP to build a better SD model.

If SD is not appropriate then:



18.6 The Role of thinking

One clarification prevents a predictable misreading of everything above. A teacher once told me that she *used DSRP on Tuesdays, in math, but not on Thursdays, in English*. I asked her to say the sentence again, replacing the word *DSRP* with the word *thinking*.

She heard it before she reached *Thursdays*.

She had not made an error. She had been handed DSRP as a noun — one more thing to take out of the cupboard on the days it was scheduled — and she was using it exactly as she had been taught. The mistake was upstream of her.

Because what she was doing on Thursday was thinking. She was distinguishing things from what they are not, assembling parts into wholes, relating them, and taking Perspectives on them. The same four Patterns she had been told were a Tuesday activity.

The notation of Section 2 separates the two things her sentence had run together. T is the act of thinking — a verb, something a person does. $\mathbb{M}$ is what that act produces — a noun, the mental model or meaning that results. The relation between them (and its relation to the way the universe organizes) is the whole of O-Theory:

$$T : \quad I \xrightarrow[\text{organizes}]{O_{DSRP-484}} \mathbb{M} = I \cdot O_{DSRP-484}$$

Thinking is the act in which organization organizes information into meaning. DSRP-484 is a description of the O in that expression.

Which is why the Thursday sentence cannot be completed. What she could decline on Thursday was the name, not the thing. The *O* term was running in her English class exactly as it ran in her math class, because *T* does not proceed without it. Declining DSRP does not take the organization out of her thinking; it only leaves it unexamined. You can put down a noun. You cannot put down a verb you are in the middle of.

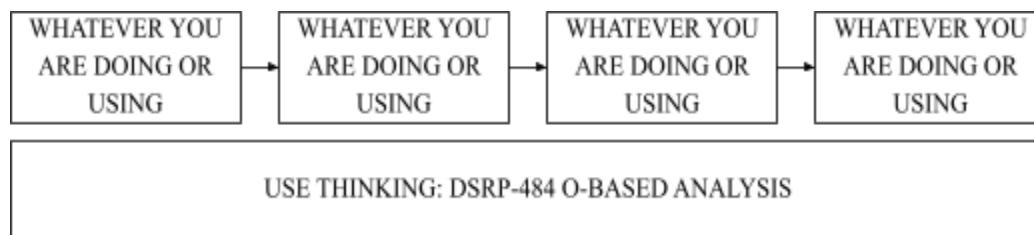
And unexamined organization is what this paper has been about from the start: the boundary drawn without noticing, the Distinction never made, the Perspective never taken, the model that runs beautifully on the wrong thing.

This is also why DSRP-484 does not care what you use. Use system dynamics. Use agent-based modeling, network analysis, ethnography, a whiteboard, a conversation, or nothing at all. Those methods were built to help people think better about particular kinds of problems, and the good ones do exactly that. But adopting a method has never been a substitute for thinking, and no framework has ever relieved its user of the obligation to keep doing it.

What is offered here, then, is not a rival framework competing for the same slot. It is the science of organization — an onto-epistemic account of how anything comes to be organized at all — and it applies to anything and everything, including whatever method is already in hand.

DSRP-484 is not a framework. It is *thinking*. Which is why it can be brought to anything, and why bringing it to something does not displace what is already there.

This yields a simple practical rule. Which is the general case:



19. The Same Test Can Unify Systems Methods

If the present decomposition is correct, the same structural-remainder test can be applied to other systems methods — and it has since been applied to network science and to UML, with the same result. What follows is the general form of the program.

A network is composed of nodes and edges. Are nodes and edges elemental, or are both recursively structured DSRP objects? A Bayesian network contains variables, conditional Relationships, graph structure, and probability distributions. Does it introduce another kind of organization? A process model contains states and transitions. Does either require an organizational primitive outside DSRP? UML contains objects, classes, inheritance, composition, association, and multiple diagrammatic views. Are these new structures or domain-specific composites?

The research program thereby shifts from asking: *Can DSRP be applied to method X?*

to asking: *Does method X contain any irreducible organization that DSRP-484 cannot generate?*

That is a far more demanding question.

As with the SD test, the candidate set for each method must be restricted to the organizational constructs distinctive or constitutive of that method. Mathematical, computational, statistical, or other machinery merely used by the method belongs to a different test. The question is what organization the method itself contributes.

This approach supplies a common basis for comparing systems methods without flattening their differences. Their specialized constructs can remain intact at the methodological level while being compared at the more elemental level of organization.

If repeated structural-remainder tests across systems methods likewise yield no residual organizational primitive, the result would be a genuinely unified systems science: not one in which all methods are declared equivalent, but one in which their differences can be expressed as different compositions of the same underlying organizational grammar.

20. Conclusion: System Dynamics Is Made of DSRP

Sixty constructs drawn from system dynamics' own account of itself were decomposed under a rule fixed in advance. All sixty decomposed. No residual organizational primitive was found. The full instrument—rule, rulings, proposed decompositions, and exclusion log—is available upon request as supplementary material. The organizational space distinctive to system dynamics lies inside the space DSRP-484 generates, and does not fill it.

Four things follow.

System dynamics is not diminished by this. Its constructs are composites, and composites are not lesser objects. Molecules remain indispensable after the elements are named. What changes is the level at which the field's structures can be explained, compared, and taught.

The constructs come with a restriction attached. To be simulable, a phenomenon must be representable as a population changing through flows under feedback. That restriction is what makes quantitative system dynamics rigorous, and it is also what makes it narrow. It applies to a system under a perspective, not to a system as such, which is why it is nearly invisible to a modeler who has already taken the perspective.

The organizational layer is doing more of the work than the field's self-description acknowledges. Most system dynamics practice never reaches a solver. What practitioners use most is the mapping, and mapping is organizational thinking conducted with one structure named and three left implicit.

The practical rule is therefore simple. Determine the organization first. If what you find is a population changing through flows, system dynamics and simulation are the right instruments, and no substitute for them is offered here. If it is not, the stock-and-flow frame will constrain what you can see before you have seen it.

The conceptual reversal can now be stated directly. The question was never whether DSRP can also do feedback, stocks, flows and delays. The question is whether feedback, stocks, flows and delays were DSRP structures all along.

On the present evidence, that is not an annexation but an inheritance. Chemists compounded and used molecules long before Mendeleev's table named the elements — and the table's empty cells then predicted gallium and germanium, later found with the properties the structure required. Forrester's constructs can likewise be understood as powerful higher-order compounds developed before their more elemental organizational basis was identified.

System dynamics remains system dynamics. But its organization appears to be DSRP.

References

1. Cabrera D, Silberman D, Cabrera L. Popularity Without Proof: A Systematic Empirical Review of Systems Thinking Frameworks. *Journal of Systems Thinking*. 2025;5: 1–29.
2. Forrester JW. *Industrial Dynamics*. Cambridge, MA: MIT Press; 1961.
3. Richardson GP. *Feedback Thought in Social Science and Systems Theory*. Philadelphia, PA: University of Philadelphia Press; 1991.
4. Sterman J. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. NEED TO FIND CITY: McGraw-Hill/Irwin; 2000.

5. Sweeney LB, Sterman JD. Bathtub dynamics: initial results of a systems thinking inventory. *System Dynamics Review*. 2000;16: 249–286.
6. Cabrera D. The unified calculus of organization, structure, emergence, and meaning: The patterns that connect mind and nature. *Journal of Systems Thinking*. 2025;5: 1–41.
7. Cabrera D, Cabrera L. DSRP-484: Building an Advanced Understanding of O-Theory, DSRP Theory, and the DSRP Calculus. *Journal of Systems Thinking*. 2026;6: 1–14.
8. Cabrera D, Cabrera L. O-Theory: The Science of Organization. *Journal of Systems Thinking*. 2025;5: 1–12.
9. Cabrera D, Cabrera L. Scale-free structure: Is knowledge just the recurring emergence of universal structures across scale? *Journal of Systems Thinking*. 2026;6: 1–40.
10. Cabrera D, Cabrera L, Cabrera E. Living DSRP Evidence Compendium for the Science of Organization. Cabrera Research Lab; 2026. Available: <https://dsrpevidence.org/evidence>
11. Cabrera D, Cabrera L. Measuring Context as Structural Organization. *Journal of Systems Thinking*. 2026;6. doi:10.54120/jost.v25i1.dc6mcnl6
12. Kant I. *Critique of Pure Reason*. 1781/1787.
13. Reichenbach H. *The Direction of Time*. Berkeley: University of California Press; 1956.
14. Bombelli L, Lee J, Meyer D, Sorkin RD. Space-Time as a Causal Set. *Physical Review Letters*. 1987;59: 521–524.
15. Richardson GP. Reflections on the Foundations of System Dynamics. *System Dynamics Review*. 2011;27: 219–243.
16. Cabrera D. Hidden in plain science: Testing a universal ontology in the results of 13 scientific disciplines. *Journal of Systems Thinking*. 2026;6: 1–24.
17. Cabrera D, Cabrera L. The Structure of Thought and Reality: Empirical Evidence for the Three Universal Laws Linking Mind, Matter, and Meaning. *Journal of Systems Thinking*. 2025;5: 1–20.
18. Ford DN. A System Dynamics Glossary. *System Dynamics Review*. 2019;35: 369–379.
19. Forrester JW. System Dynamics---The Next Fifty Years. *System Dynamics Review*. 2007;23: 359–370.
20. Coyle RG. Qualitative and quantitative modelling in system dynamics: some research questions. *System Dynamics Review*. 2000;16: 225–244.
21. Homer J, Oliva R. Maps and models in system dynamics: a response to Coyle. *System Dynamics Review*. 2001;17: 347–355.
22. Stave K, Zimmermann N, Kim H. Qualitative Aspects of System Dynamics Modeling. *System Dynamics Review*. 2024;40: e1793.
23. Caddell JD, Nilchiani R. The Dynamics of Trust: Path Dependence in Interpersonal Trust. *IEEE Engineering Management Review*. 2023;51: 148–165.