
Descriptor Objects: The First Pillar of Systems Theory as a Co-Creation Engine

Vidyadhar Tilak

From Meaning to Model, and from Model to Action

Abstract

Systems theory, when interpreted as a co-creation engine, rests on three irreducible pillars: descriptor objects, real-world objects, and the processes of co-creation that bind them through modelling and simulation. This essay introduces descriptor objects as the foundational pillar that enables systems to be conceived, represented, reasoned about, and simulated prior to intervention in the real world. Descriptor objects are not passive notations but active cognitive and computational instruments that mediate between human intention, formal structure, and executable models. We argue that descriptor objects naturally organise into three interdependent domains—Basic Mathematics, Core Mathematics, and Linguistics—each contributing a distinct but inseparable role in systems modelling. Together, they form the symbolic and structural substrate that makes "test before change" both possible and meaningful.

1. Introduction:

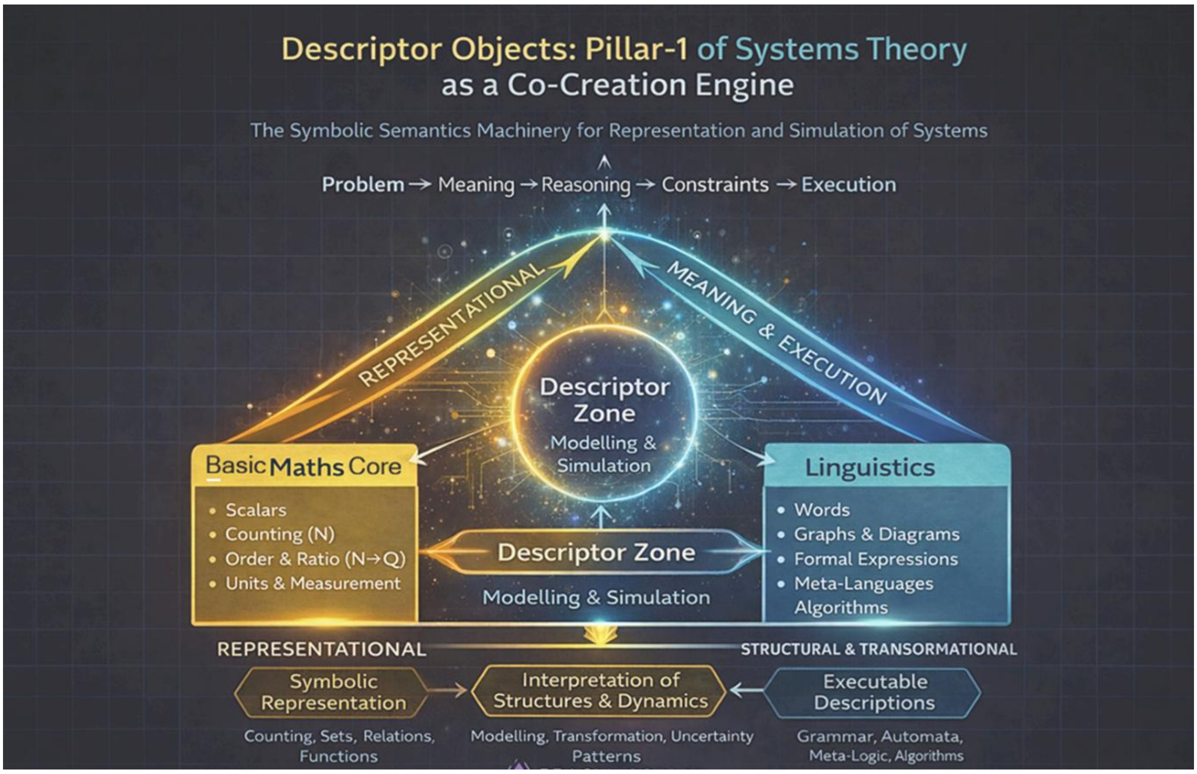

Figure 1: Descriptor Objects The first Pillar of Systems Theory

No system is accessed directly. Whether the system is physical, biological, social, or socio-technical, it enters human cognition and collective action only through *descriptions*. Systems theory, therefore, must begin not with the world as it is, but with the means by which the world is *represented*. Descriptor objects constitute this means. They precede modelling, simulation, prediction, and validation. Without them, the notion of co-creation collapses into trial-and-error interaction with reality.

Descriptor objects are the formal and semi-formal constructs through which structure, behaviour, constraints, and change are expressed. They enable the transition from amorphous experience to 'articulated' knowledge, and from 'articulated' knowledge to executable models. In this sense, descriptor objects are not merely tools of analysis; they are instruments of *responsible intervention*.

In the co-creation framework, descriptor objects occupy a unique epistemic position. They stand between real-world objects and the processes that transform understanding into action. Real-world objects—energy, matter, organisms, institutions—possess intrinsic dynamics independent of our representations. Co-creation processes—design, simulation, optimisation, governance—operate on representations, not directly on reality. Descriptor objects form the bridge.

This mediation role demands that descriptor objects satisfy three conditions simultaneously. First, they must be *conceptually expressive*, allowing systems to be imagined and reasoned about. Second, they must be *formally precise*, enabling consistency, proof, and reproducibility. Third, they must be *computationally executable*, allowing simulation, exploration, and predictive testing. The organisation of descriptor objects must therefore reflect these demands.

2. The Triadic Structure of Descriptor Objects

Descriptor objects organise naturally into three mutually reinforcing domains: Basic Mathematics, Core Mathematics, and Linguistics. This is not a curricular convenience but a systems necessity. See Figure 2 Below.

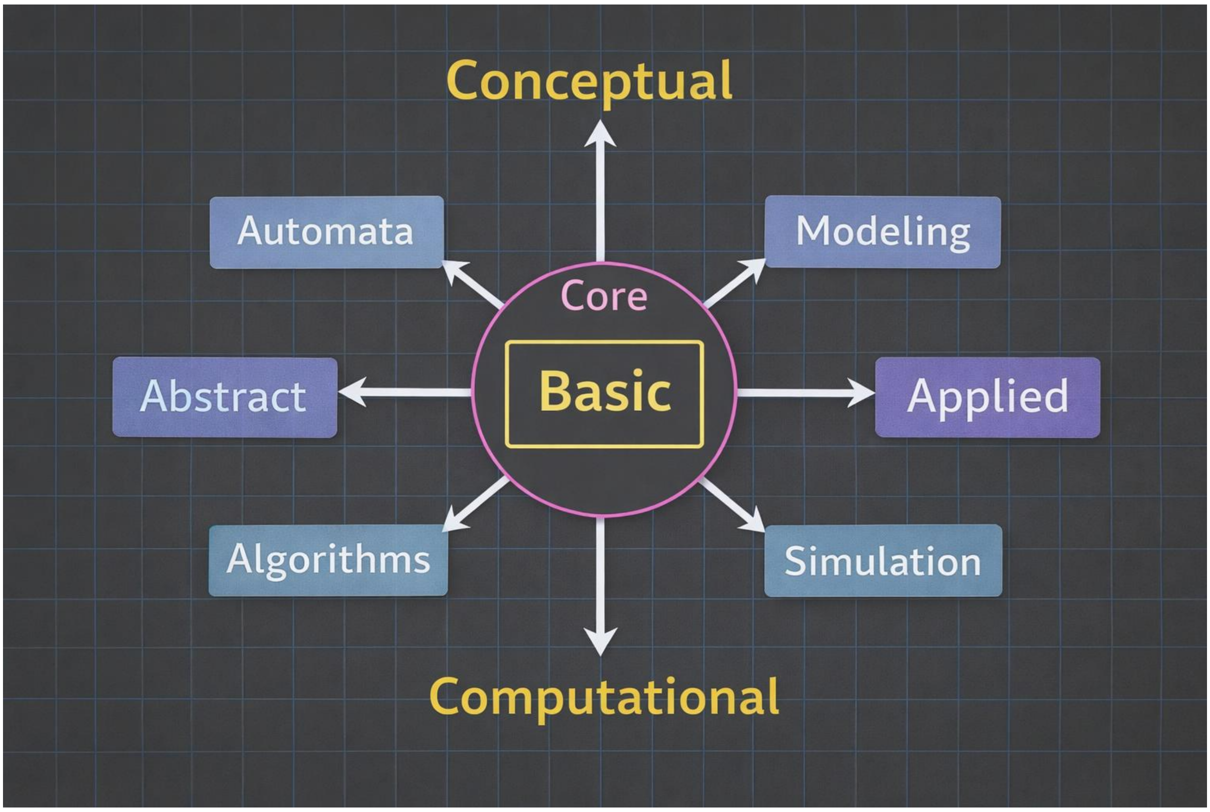

Figure2: Components of Descriptor Objects

Basic Mathematics provides the primitive symbolic infrastructure. Core Mathematics provides structural and transformational depth. Linguistics provides meaning, articulation, and executability. None can function independently in systems modelling.

The figure 3 below, showing arithmetic at the core, logic and set theory as scaffolding, and algorithms and simulation as outward extensions—visually reinforce this triadic organisation.

3. Basic Mathematics: The Primitive Descriptor Layer

Basic Mathematics supplies the *atomic descriptors* of systems. It includes arithmetic, counting, ordering, ratios, magnitudes, and elementary logical distinctions. At this level, numbers are not abstract entities but instruments of measurement and comparison. Units, scales, and proportional reasoning belong here.

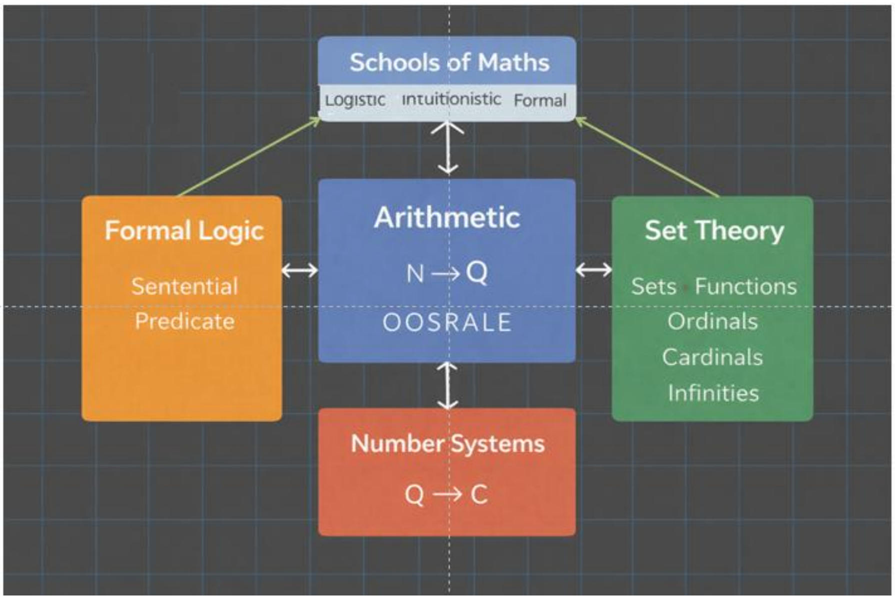

Figure 3: Basic Mathematics

Crucially, Basic Mathematics introduces discreteness and repeatability. Counting presupposes distinguishable entities; ordering presupposes comparability; ratios presuppose invariance across contexts. These assumptions are minimal, yet they underpin every higher-order system description. In systems terms, Basic Mathematics enables *object individuation* and *property attribution*. Without it, neither structure nor change can be stabilised.

4. Core Mathematics: Structural and Transformational Descriptors

Core Mathematics extends basic descriptors into structured systems capable of expressing relations, patterns, and dynamics. This domain includes algebra, geometry, calculus, probability, and statistics, unified not by topic but by function.

Algebra provides symbolic structure and rule-based transformation. Geometry and topology encode spatial and relational organisation. Calculus captures temporal evolution and continuous change. Probability and statistics formalise uncertainty, variability, and macro-patterns emerging from micro-interactions.

The unifying insight here is that Core Mathematics transforms static descriptions into *relational and dynamical models*. Forces become vectors, oscillations become functions, fields become tensors, and uncertainty becomes distributions. Systems cease to be collections of properties and become networks of interactions evolving in time. See Figure 4 below.

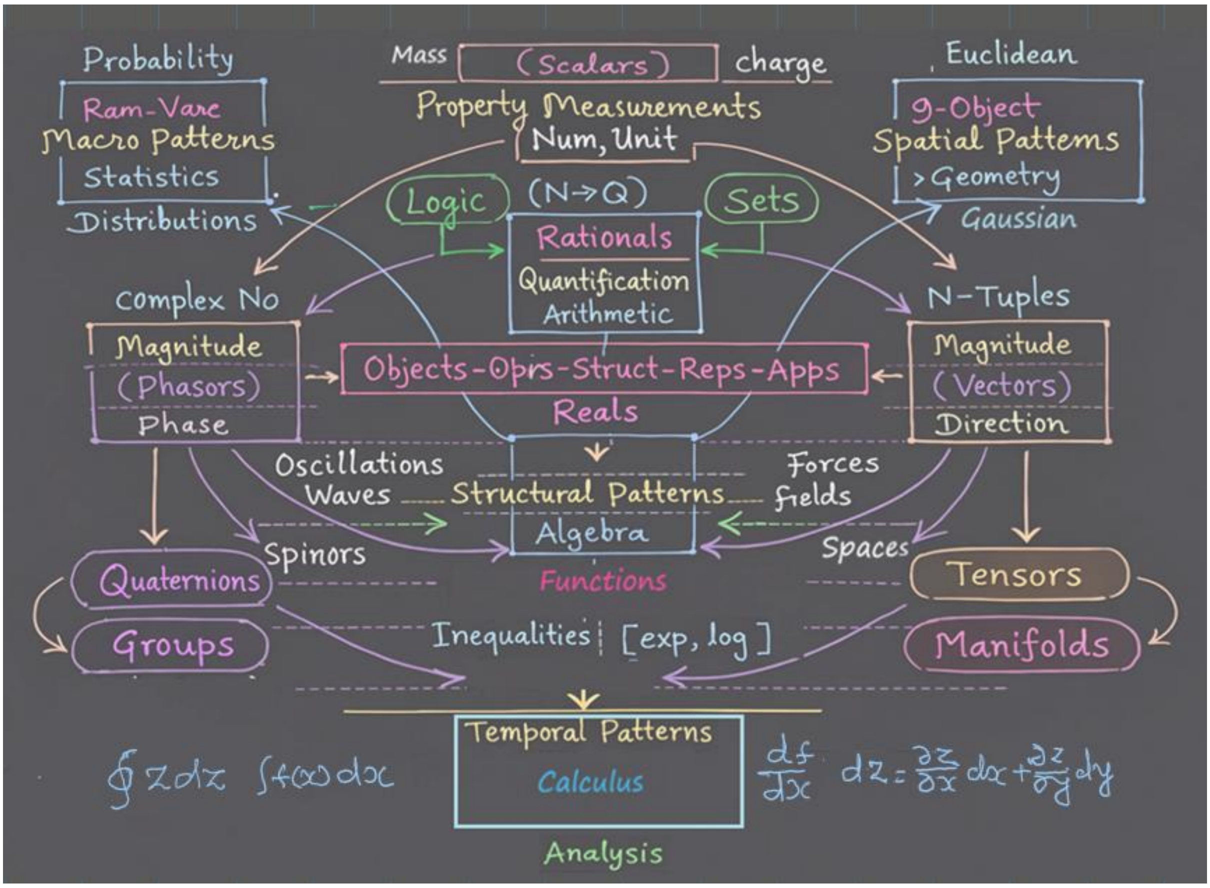

Figure 4: The Core Mathematics

5. Linguistics: The Meaning–Execution Substrate

Linguistics, in this framework, must be understood in a systems-theoretic sense, not merely as the study of natural language. Linguistics is the descriptor domain that enables symbols to *mean*, *refer*, and *act*.

At its foundation lie symbolic representations: words, numbers, graphs, diagrams, and expressions. These are governed by syntactic rules—grammars, formalisms, and compositional constraints—that define what constitutes a valid description. Beyond syntax lies semantics: the mapping between symbols and the world, between equations and phenomena, between models and systems.

At the highest layer lies pragmatics and execution. Here appear meta-languages, formal specification languages, algorithms, automata, and processors. The classic triangle of **Problem–Language–Processor** finds its natural home here. Problems articulate intent, languages formalise intent, and processors enact intent through computation or simulation. See Figure 5 Below.

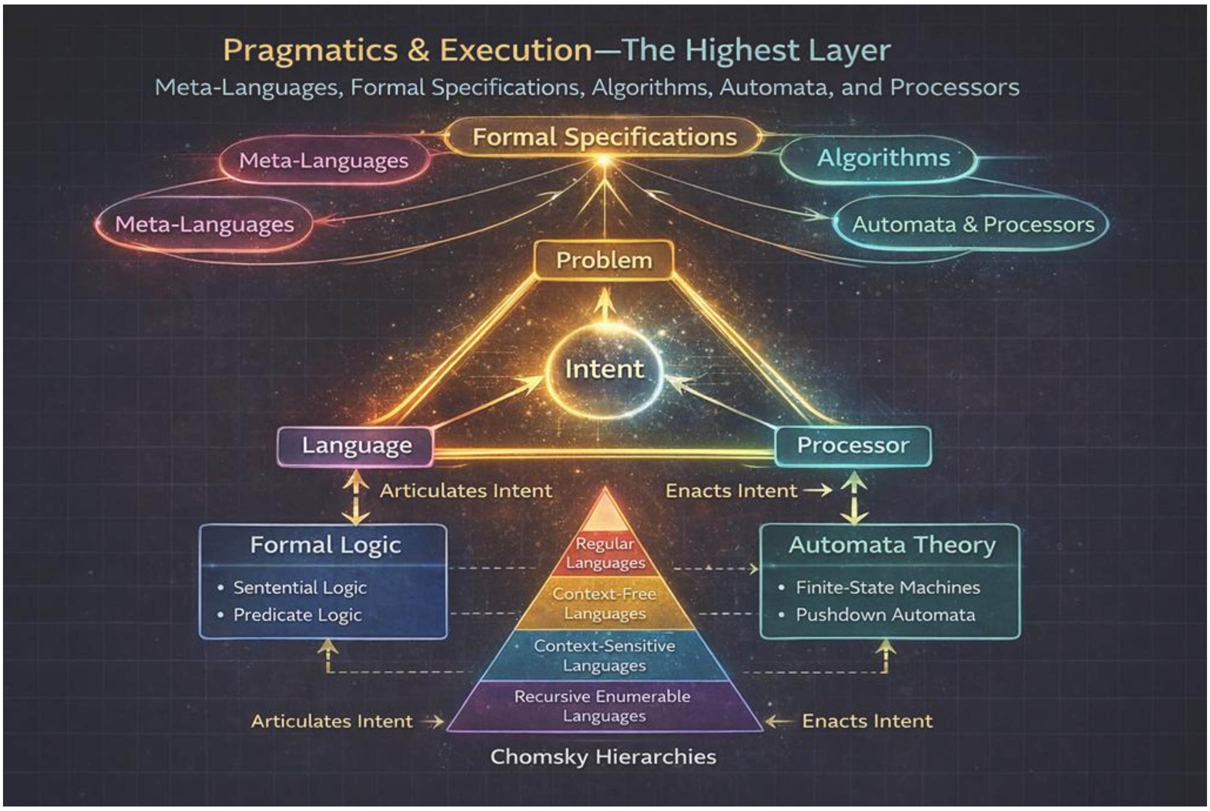

Figure 5: Pragmatics & Execution

Chomsky hierarchies, formal logic, and automata theory are not abstract computer science topics in this view; they are **co-creation enablers** that determine what kinds of systems can be described, simulated, and governed.

6. Descriptor Objects and "Test Before Change"

The central ethical and practical promise of systems theory as co-creation lies in the principle of *test before change*. Descriptor objects make this possible. Mathematics without linguistics remains inert symbolism. Linguistics without mathematics lacks structural discipline. Together, they produce executable models capable of simulation.

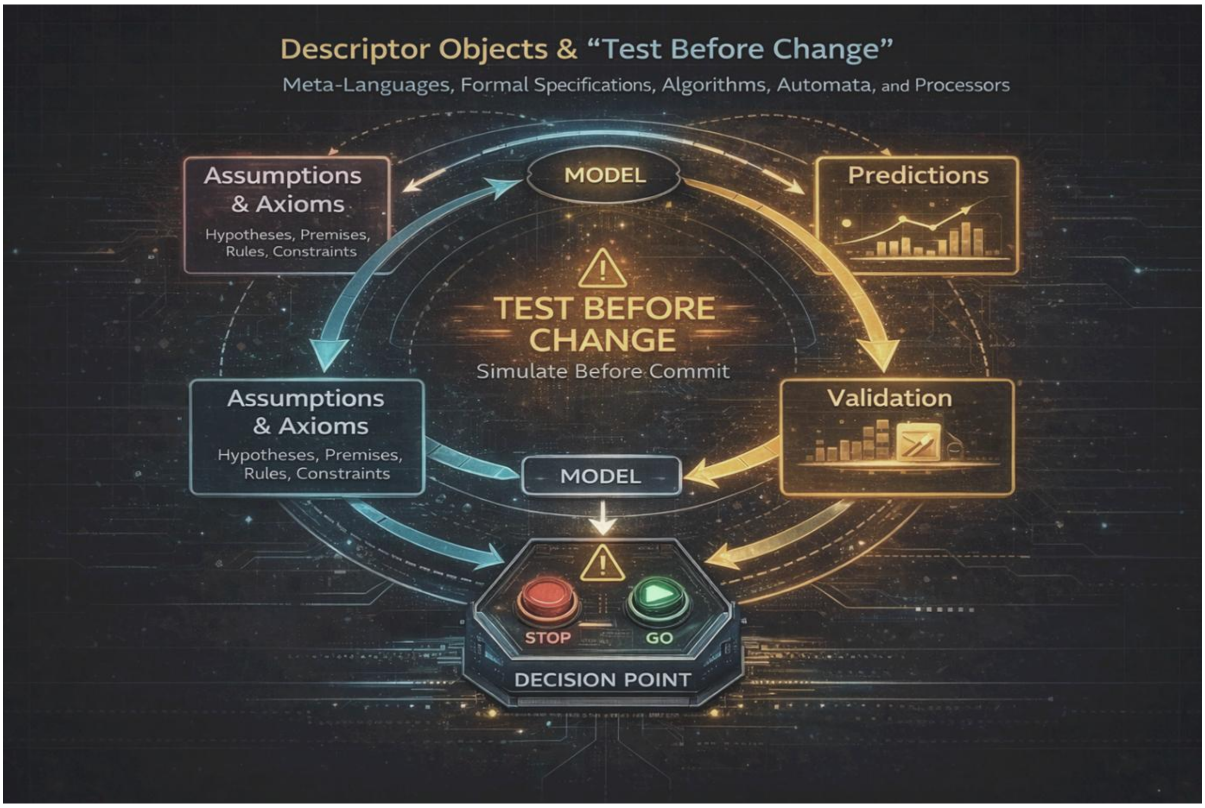

Figure 6 Descriptor Objects-Test before Change

Simulation is not an afterthought; it is the culmination of descriptor object design. It is here that predictions emerge, assumptions are stressed, and unintended consequences are surfaced before real-world commitment. The figures illustrating the flow from hypotheses and axioms to predictions, proofs, and validations capture this feedback loop with precision.

An important but often neglected insight is that descriptor objects are not merely external tools; they shape cognition itself. What can be described can be thought. What can be modelled can be debated. What can be simulated can be responsibly changed.

In this sense, descriptor objects form a civilisation's *cognitive infrastructure*. The evolution from arithmetic to calculus, from natural language to formal languages, and from static models to executable simulations mirrors the evolution of humanity's capacity for collective co-creation.

7. Positioning Descriptor Objects within the Three Pillars

With this foundation, the role of descriptor objects within the three-pillar architecture becomes clear. Descriptor objects define the representational space. Real-world objects define the ontological space. Co-creation processes—modelling, simulation, design, governance—define the transformational space.

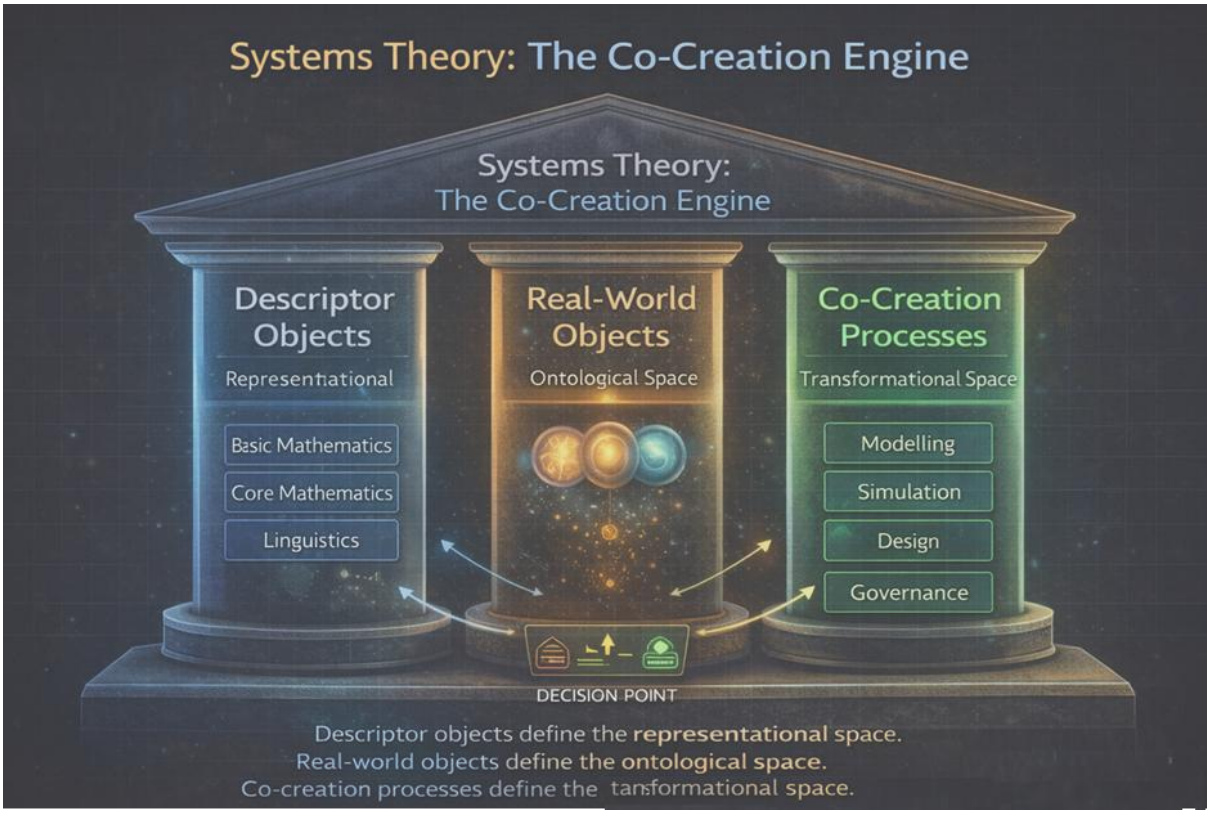

Figure 7: Positioning Descriptor Objects within the Three Pillars

The pillars are sequential but inseparable. Descriptor objects come first not because they are superior, but because without them, nothing else can proceed coherently.

8. Recapitulation

Descriptor objects constitute the first and foundational pillar of systems theory as a co-creation engine. Organised into Basic Mathematics, Core Mathematics, and Linguistics, they provide the symbolic, structural, and executable means by which systems are conceived, analysed, and responsibly transformed. They enable the critical shift from direct intervention to mediated co-creation through modelling and simulation. In doing so, they convert systems theory from a descriptive discipline into an ethical and practical instrument for shaping the future.

From Meaning to Model, and from Model to Action

------------- 0-------------

References

Rosen, R. *Anticipatory Systems*. Rosen's *Anticipatory Systems* argues that living systems differ from machines because they contain internal models that let them act based on predicted future states. He formalizes this through the modeling relation and shows that anticipation requires organizational closure that mechanistic systems lack. The book establishes anticipation as a defining property of life, reshapes systems theory, and provides a relational, non-reductionist foundation for understanding organisms, cognition, and complex adaptive systems.

Simon, H. *The Sciences of the Artificial*. Herbert Simon's *The Sciences of the Artificial* argues that human-made systems—defined by goals, functions, and adaptation—are as worthy of scientific study as natural systems. He positions design as a universal, interdisciplinary process of transforming situations into preferred outcomes, and explores how complexity arises from simple interactions within adaptive environments. The book laid foundational ideas for cognitive science, artificial intelligence, and systems theory, reshaping how we understand problem-solving, decision-making, and the nature of artificial constructs.

Chomsky, N. *Syntactic Structures*. Noam Chomsky's *Syntactic Structures* introduced transformational-generative grammar, a formal system that generates all grammatical sentences of a language using recursive rules. It argued that syntax is independent of meaning, that humans possess innate linguistic capacities, and that language can be modeled mathematically. The book launched the cognitive revolution, reshaping linguistics into a formal science and influencing psychology, artificial intelligence, and philosophy.

Zeigler, B. *Theory of Modelling and Simulation*. Zeigler's *Theory of Modeling and Simulation* establishes a rigorous scientific foundation for modeling and simulation, centered on the Modeling Triad (real system, model, simulator) and the DEVS formalism. It provides a mathematically precise, modular, and hierarchical framework for representing dynamic

systems and executing them through simulation. By introducing concepts like experimental frames, closure under coupling, and formal system specifications, Zeigler unifies diverse modeling paradigms and defines simulation as a disciplined scientific activity rather than an ad hoc engineering practice.

Churchman, C. W. *The Systems Approach*. Churchman's *The Systems Approach* argues that systems thinking is not a neutral technical method but a deeply ethical and philosophical practice. Every system is defined by choices about boundaries, goals, and values, and these choices inevitably affect people. Churchman identifies the "enemies" of the systems approach—politics, morality, religion, aesthetics, and knowledge—not as obstacles but as essential dimensions of inquiry. He calls for a reflective, ethical, and inclusive systems practice that acknowledges uncertainty, confronts assumptions, and **seeks idealized systems that consider all stakeholders.**

Technical Terms

Term	Description
Algorithm	A finite, well-defined sequence of executable instructions designed to transform inputs into outputs within a system model.
Arithmetic	The foundational mathematical system concerned with counting, ordering, ratios, and basic operations enabling measurement and comparison.
Automata	Abstract machines defined by states and transition rules that interpret and execute formal descriptions.
Basic Mathematics	Primitive symbolic constructs such as numbers, order, ratios, units, and elementary logic used for system description.
Calculus	Mathematical framework for describing continuous change, accumulation, and dynamic system behaviour.
Chomsky Hierarchy	Classification of formal languages based on their generative and syntactic power.
Constraint	A formal restriction on system variables or behaviour defining admissible states.
Core Mathematics	Algebra, geometry, calculus, probability, and statistics providing structural and transformational descriptors.
Descriptor Objects	Symbolic, formal constructs used to represent, reason about, and simulate systems.
Executable Description	A formal representation that can be directly interpreted by a processor or automaton.
Formal Language	A symbol system with explicit syntax and semantics free from contextual ambiguity.
Formal Logic	Rule-based symbolic system enabling valid inference and proof.
Function	A mapping assigning a unique output to each input, expressing system dependencies.
Grammar	Rules governing valid composition of symbols into expressions.

Graph	A relational representation consisting of nodes and edges encoding system connectivity.
Interpretation	Mapping of formal expressions to meanings or real-world referents.
Language (Systemic)	Structured symbol system enabling representation, communication, and execution of meaning.
Linguistics (Systems-Theoretic)	Descriptor domain concerned with representation, syntax, semantics, and execution.
Meta-Language	A language used to define or analyse another language.
Model	A structured representation capturing relevant properties and dynamics of a system.
Modelling	The process of constructing formal system representations.
Natural Language	Human language used primarily for conceptualisation and intent articulation.
Probability	Mathematical framework for quantifying uncertainty and variability.
Problem Statement	Formal articulation of intent, objectives, and constraints.
Processor	Mechanism that interprets and executes formal descriptions.
Representation	Symbolic depiction of objects, properties, or relations.
Semantics	Study of meaning and reference of symbols and expressions.
Simulation	Execution of a model to explore behaviour without real-world intervention.
Specification	Precise formal description of system requirements or behaviour.
Statistics	Mathematical discipline dealing with data, variability, and inference.
Syntax	Rules governing structural validity of expressions.
System	An organised set of interacting components exhibiting collective behaviour.
Test Before Change	Principle of validating interventions via modelling and simulation.
Transformation	Rule-governed change applied to system states or representations.
Unit	Standard reference for measurement anchoring quantities to reality.