Systems Ontology

The Competence, Its Ground, and Its Practice

P. Orelio Sattari · Version 1.0, November 15, 2026

I. What a system is

A system is a collection of interconnected parts that work together as a whole to achieve a purpose or function.

There are three things present in any system: the parts, the relations between the parts, and the *whole*—the organizing principle or purpose around which the parts and relations cohere. These three are not equal in causal standing. The whole comes first, because the purpose is what makes the parts parts at all. The same tissue that is a heart within a circulatory system is, removed from it, only matter. What makes it a heart is its role in the whole. Before something can be divided into parts, there must be something to divide. The parts are derived, and the whole is the ground of their derivation.

The relations are as consequential as the parts, with properties of their own that belong to neither of the things they connect. Two gears meshed together have a ratio. The ratio is not a property of the first gear or of the second—it exists only in their engagement, and it determines what the pair can do. This is why two components can each be well made and badly matched, and why a system can be assembled entirely from sound parts and still fail. The fault may lie in a relation, which no examination of the parts will disclose. Relations are also the most commonly unrecorded feature of built systems. They exist wherever two parts affect each other—through shared inputs, sequence, dependency, or contention for the same resource—whether or not anyone has written them down. And a relation that operates without being documented is still doing its work.

The whole, in turn, has properties that belong to none of its parts. A mechanical clock is made of gears, springs, and escapements, but no gear tells the time. Telling the time is a property of the assembly, present only when the parts stand in the right relations, and absent from every component considered alone. Properties of this kind are not mysterious, but they are systematically invisible to any method that proceeds by examining components alone—which matters, because they are frequently the properties the system exists for.

And finally, systems can *nest*. A part, examined on its own terms, is usually a system in its own right. The heart has chambers, valves, muscle, and a conduction system, arranged for a purpose and coherent as a whole. The system that heart belongs to is itself a part of something larger. This means that "what counts as the whole here?" is a real question rather than an obvious one, and answering it is the first move in understanding anything. Read a

hospital ward as the whole and you will see one set of things. Read the patient's *path through the hospital* as the whole, with the ward as one part of it, and you will see another.

Which raises the question of where a system ends. Every reading of a system requires drawing a boundary, and the boundary is a judgment. Consider a hospital emergency department that is failing to meet its waiting times. Draw the boundary at the department and you will study triage, staffing, and throughput. But the department's capacity to move patients out depends on bed availability elsewhere in the hospital, which depends on discharge processes, which themselves depend on services outside the hospital altogether. If the binding constraint lies beyond where the boundary is drawn, then every improvement made inside of it will be real and measurable while still being insufficient—and the failure will be attributed to the department, because that is where the boundary was drawn.

The same holds for how a system is divided inside that boundary. A hospital can be cut into parts by ward, by profession, by medical condition, or by the path a patient travels from admission to discharge. Each of these is a real way to divide it, and each is in use somewhere. But they are not interchangeable, because a division decides what can be seen. Cut the hospital into wards and each ward can be staffed, measured, and held to account—while a patient who deteriorates during the handover from one ward to the next has been harmed in a place that is no ward's responsibility, because the harm happened between the parts the division itself created. The same event is invisible cut one way and obvious cut another. This means a division can be done well or badly, and the test is whether it keeps together what actually works together. A hospital divided along the patient's path keeps each handover inside a single part where someone owns it. A hospital divided by ward splits that same handover down the middle and assigns it to no one. Neither division is false. One of them can see the patient who was lost in the gap, and the other cannot.

Systems exist at every level of reality. An atom is a system. A cell is a system. A national grid, a firm's order fulfilment, a court's rules of evidence, a scientific theory—each is a collection of interconnected parts working together for a purpose. Each can be understood the same way: by grasping what it is for, what composes it, how the components relate, and how the whole holds it all together. If we zoom out to the broadest, most general level, we arrive at nature itself, coherent and patterned from fields to organisms, and why understanding gained at one level carries to another.

II. What ontology is

Ontology is the study of what is, and of the kinds into which what is divides.

Those sound like two subjects and are one. Ask what exists and you cannot answer for long in the singular. The world does not present itself as one undifferentiated thing but as an immense variety of things standing in relations, and to say what is, you must say what kinds there are and how they stand in relation to each other. The question of ‘being’ generates the

work of categories. The work of categories is what the question of being looks like when it is actively pursued rather than merely posed.

This has a consequence that is easy to miss. Categories can be right or wrong. Not right or wrong in the sense of useful or useless for some purpose, but right or wrong about the world. A category asserts that a certain group of things belong together and are distinct from others, and that is either true of the thing being categorized or it is not. This is why ontology is not simply a matter of definition or preference. Definitions can be stipulated and we may use a word however we like. But categories are not stipulations. They assert something about how the world is jointed, and the world either has those joints or it does not.

Today, the word ontology is used in two ways that are usually presented as unrelated. In philosophy it names the study of being—the question of what is real. In information science it names a formal specification of the categories in some domain and the relations among them, built so that different systems can exchange the same types of information without misunderstanding one another. A philosopher and a data architect who both say they work on ontology are doing similar activities, but will not, at present, recognize each other's work.

That separation is recent and was made somewhat deliberately. Until less than half a century ago, there was no split to speak of. Aristotle's *Categories* is at once a claim about the fundamental kinds of being and a scheme for organizing everything sayable about a subject. The same is true of Ibn Sīnā dividing being into the necessary and the contingent. The category work and the being question were the same work, because a category scheme was understood to be answerable to reality. The point of drawing the categories correctly was that reality was already divided some way, and one's scheme should follow it.

What separated them was the arrival of a practical need for schemes that did not have to be true. When computing systems had to exchange information, they required agreed vocabularies—a shared specification of what counts as a customer, an event, etc. Whether that specification described the world correctly was not the operative question; whether the systems could inter-operate was. The word was borrowed for this work in the early 1990s, and the borrowing carried the category scheme away from the question of truth. Philosophy kept the being question and mostly relinquished the practical category work. Neither half is wrong about its own concerns, but the separation removed the discipline that had connected them—the practice of drawing categories and being answerable for whether they were right.

Ontology can be practiced on anything, because everything that exists has some structure, and structure is its object. This is not a claim to expertise in every domain. A specialist has deep knowledge of what is true within their field. But an ontologist asks whether the field's categories are drawn where its subject actually divides. Those are different questions, and answering the second does not require having answered all of the first.

A systems ontologist, then, is not someone applying a philosophical discipline to a practical subject. It is the name for what ontology has always been when it is pointed at things that

were built—the reading of what a thing is, what it is composed of, how its parts stand to one another, and whether the divisions it runs on correspond to what is actually there.

III. Where it sits among other systems disciplines

Anyone who has worked near systems will have noticed that the ground described so far is not totally empty. Several fields take systems as their subject, and each works on one aspect of what a system is.

Systems engineering builds a system to meet requirements. It takes a specified purpose, decomposes it into components and interfaces, and manages their integration so the assembled thing performs. Its subject is construction—the artifact being brought into being to a particular specification.

Cybernetics and control theory study regulation—how a system holds a variable steady against disturbance, how feedback returns it toward a target. Control theory is the mathematical core. Cybernetics generalized it across machines, organisms, and societies. Their subject is how a system stays on course.

System dynamics studies how a system behaves over time. Stocks accumulate, flows run between them, delays and feedback loops produce growth, collapse, and oscillation that no component predicts on its own. Its subject is behavior through time.

Complexity science studies emergence—how order at the scale of the whole arises from simple rules followed by many parts, none of which holds the pattern alone. Its subject is how the large-scale order comes from the small-scale behavior.

Enterprise architecture maps and aligns the systems of an organization—its processes, data, applications, and technology—so they fit together and serve the business. Its subject is bound to one kind of system, the firm.

Notice what these have in common. Each takes one aspect of a system and goes deep. Engineering takes construction. Cybernetics takes regulation. Dynamics takes behavior. Complexity takes emergence. Every one of them is a specialization: a division of systems work into a slice, pursued by practitioners trained in that aspect and largely confined to it. This is the fragmentation described elsewhere in this body of work, now operating on the study of systems itself—the subject cut into aspects, and the aspects made into separate fields with separate literatures.

The specialist goes deeper. On the engineer's own ground the engineer knows more, and an ontologist who claimed otherwise would simply be wrong. But depth in an aspect and command of the whole are different competences, and it is the second that goes missing when a field is divided into specialties—because the division that creates the specialties is the one thing that none of them is left holding and that none of them is equipped to see.

Those disciplines take an *aspect* of systems as their object. Ontology takes what a system *is* as its object—the parts, the relations, the whole they compose, and whether the divisions in play correspond to what is actually there. That object is not a region of systems work. It is the thing all of systems work is about, which is why it does not sit alongside the others as a peer but underneath them as a floor.

One field set out to be general in something like this way. General Systems Theory (GST) looked for the principles common to all systems. Austrian scholar Von Bertalanffy's work searched for what holds stable whether the system is an organism, a machine, or a society—feedback, hierarchy, open exchange with an environment, the reaching of one end by many paths. It is a generality, but it is a different one.

General Systems Theory looked for the laws systems share—findings about what is common to systems, which you then carry to a particular system and apply from outside. Its subject is the regularity that recurs. Systems ontology has a different subject. Not what systems have in common, but whether this system—the one in front of you—is constituted as it should be. That is not a law and cannot be derived from laws. You could know every regularity systems obey and still not see that a particular hospital is divided where no patient is, because that is not a general feature of systems—it is a joint in one specific thing, found only by reading that thing against what it is for. General Systems Theory is a science of what systems have in common. Systems ontology is the reading of whether one system is rightly made.

IV. Built systems, actuality, and the correction loop

Every built system begins as an idea—a design, an intended arrangement, a conception of what the thing will do—and then operates in the actual world, which is always messier than the conception. The design lives in the register of the "ideal." The operation lives in the register of the "actual." Between them there is always a gap, because the ideal is always cleaner than the actual. A system's adequacy is therefore not a property it has once, at birth. It is something maintained.

What maintains it is *feedback*. Where a system's workings are tested against actuality—where mistakes produce consequences that flow back quickly and visibly—the arrangement gets corrected and adequacy is preserved. A clear example of this in nature is the nervous system. An organism's representations of its environment stay faithful because unfaithfulness is punished by the world directly and immediately, at stakes up to death. The loop never breaks, and so the map never drifts far from the territory.

But many systems, and many regions within systems, have no such loop. Their assumptions are not tested against the world at all, or are tested so slowly and diffusely that correction never arrives. And here is the question that follows from first principles, the one that can be asked of any system and is often overlooked: *in the regions where this system's understanding is not tested against actuality, what mechanism exists to correct it?* If none exists, what due

diligence has been done to determine how well those regions match reality? More often than not, the honest answer is that no such mechanism exists and no such diligence has been done. The system runs on its untested regions with the same confidence as its tested ones, and nothing inside the system distinguishes between the two.

This is where the most consequential failures of arranged systems often live—not in broken components (which announce themselves) but in untested assumptions (which do not). It is also the sharpest way to understand the newest and largest case. An AI language model is a system whose entire product is its representation of the world, and the question of what corrects those representations—and what the correctors' own incentives are—determines where its outputs can be trusted and where their reliability is simply unknown, to the system and its users alike. That case is treated fully under separate cover. Here it simply stands as one instance of the general principle. Adequacy is maintained by correction. Where correction is absent, adequacy is unverified—and unverified is not the same as sound.

V. What a systems ontologist does

The competence is one act: comprehending a system. Grasping what it is for, what parts compose it, how the parts relate, and how the whole coheres and is faring against what it operates in. The practice is that act pointed at the different situations systems present across their lives.

Designing systems. Before anything is built, the comprehension runs in its native direction. We begin from the purpose and derive what must exist to serve it: which functions the purpose actually requires, in what arrangement, at this scale, for this organization. Most design failures are not failures of construction; they are failures to derive the right arrangement from the purpose in the first place, which is why an inadequate architecture can be caught, and a better one drawn, before any materials are paid for. It is the same reason blueprints are taken for a second opinion before a house is built.

Guiding construction. While a system is being built, the arrangement can drift from the intention, quietly, through a hundred local decisions that each seem harmless. An independent eye that holds the whole picture can see what the builders, each inside their portion, cannot—where the execution has departed from what the design understood. This is what a building inspector does for houses. The same function exists for every built system, whether or not anyone has been hired to perform it.

Maintenance / Operational alignment. A running system benefits from someone who retains the full picture of the machine—what it is for, how its parts relate, and where its couplings and dependencies lie. With that picture held, the system can be kept adequate to its purpose as the organization changes around it, and when something goes wrong, the picture shows where to look.

Investigating errors and malfunctions. When something is wrong and no one can find it, the direct path is to read the system: what it is for, how it's arranged, and where in that arrangement a fault of this character can live. This also identifies the nature of the error, not just its location—whether it's an absent or misaligned purpose, a missing feedback loop, a defective component, a missing or broken relation, a flawed arrangement, a false assumption, or a division that does not correspond to where the thing actually divides. The remedy follows from the nature. Organizations that misdiagnose the nature of an error can patch its symptoms for years without actually solving the problem. This work applies anywhere: a company, a court system, an insurer, an agency, a country—any organization running an arrangement that is misbehaving.

Analyzing complete systems. A full analysis of a system can be carried out formally and its results quantified and scored. It examines the system's purpose, its parts, the relations between them, its coherence, its security and permissions, its adequacy to what it runs in. Each component receives a score, as do the relations between components and the system as a whole, weighted by the consequence of what was found and by how far upstream it sits—combined into an overall grade. A few findings are binary, but most fall on a range, and a system can score well overall while scoring poorly in particular functions.

Theoretical work. The competence also produces research: papers, case studies, and published work on systems and ontology as such, on the patterns and principles and the structure of wholes and parts across levels.

Communication and counsel. The practice includes externalizing it—teaching, advising, and articulating the principles of design and maintenance of complex systems so that others can hold and apply them.

Two things on this list are unlike the others. The first five point the competence at a system in front of you—designing it, guarding its construction, holding its picture, finding its faults, and grading its condition. The last two point it elsewhere—theoretical work studies systems as such, and counsel puts the comprehension into other hands. They are the practice turned outward, the difference between a physician treating patients and one who also writes on anatomy and teaches the residents. A discipline needs both, because a competence that is only ever exercised, and never articulated or transmitted, dies with its practitioners.

And this discipline has almost died that way many times, because it has never been carried by an institution. It has been carried by individuals—the engineer who could not stop seeing the whole plant, the physician who treated the patient rather than the chart, the administrator who understood what the org chart concealed. Every field has a few. They are the ones colleagues often bring the unsolvable problems to, the ones who seem to know where to look before anyone has explained the system to them. What they hold has typically had no name, no training path, and no way to outlive them. Each one has had to reassemble the whole competence alone, from scratch, against the grain of an education that taught them to specialize.

Systems ontology, as outlined here, is therefore not a new method, but a name for an old competence. One that—once named—can be recognized, transmitted, and held somewhere. The specialists it works alongside lose nothing. The engineer still builds, the architect still maps, the analyst still models, and each still knows their aspect more deeply than the ontologist will. What changes is that the thing between their specialties—the whole that every one of them stands inside and none of them is assigned—stops being an accident of unusual individuals and becomes someone's work.

Systems fail in the gaps between the people responsible for their parts. They are understood—when they are understood—by someone who took the whole as their object. Every discipline that studies systems knows this. It is the first thing their textbooks say and the last thing their institutions reward. Systems ontology is nothing more than the decision to take that first page seriously—to make the whole a discipline of its own, so that what has always depended on the rare individual who happened to see it can finally be asked for, taught, and practiced by design.

© 2026 Sattari Group, LLC