Cognitive Grammar in relation to Linguistics is an interesting analog to Phenomenology in relation to Philosophy. Thematically, Cognitive Grammar approaches linguistic phenomena from the perspective of language users' experience of percptual and situational structures — as such, language structure is a kind of formalization, communication, and/or re-construction of patterns of organization within consciousness. This perspective certainly seems amenable to Phenomenology more immediately than paradigms that connect language-meanings to logical propositions, or to the mechanical operations of semantic and grammatical rules. Meanwhile, methodologically, Cognitive Grammar embraces a style of analysis grounded in first-person perspective; the linguist's reflective judgments on which potential expressions in a given language would be deemed acceptability to typical speakers of that language, and why. True, that kind of subjective but evidence-based methodology — linguists using their own sense of syntactic propriety and semantic coherence to determine qualifications like the acceptability of particular sentences, but doing so in the guise of generic speakers' language-use rather than any idiosyncratic preferences the linguist herself may have — is common across many paradigms of linguistic research. In the case of Cognitive Grammar, however, the researcher imagines linguistic phenomena situated in hypothetical perceptual and interpersonal contexts. Issues like acceptability are accordingly assessed with consideration to the overall experiential, practical, and situational enacting and conceptualizing that would constitute the cognitive givens talked about and grounding communicative practice. Insofar as practical and experiential context is treated as intrinsic amongv the details needing to be consulted for trustworthy assesments of, say, acceptability — rather than comparingb a given linguistic performance against dictionaries or style-books — Cognitive Grammar eveals a metatheoretic commitmnt to the practical/experiential environing of cognition in general as something that can be systematically investigated, enough to make linguistics work as a rigorous science.

Cognitive Grammar, in short, requires for the coherence of its scientific aspirations that a kind of first-personal but social-pragmatic analysis — which we can argue is quite akin to Phenomenology — can be performed which is also scientifically rigorous and amenable to third-person follow-up; phenomenological claims being reviwed and synthesized within a scholarly community. Cognitive Grammar needs a story of its scientific merit that overlap with Phenomenology's parallel story: that first-person reports do not devolve into stream-of-consciousness but can mine experience for regulative patterns and recurring desiderata that can be critically examined by a sufficiently sympathetic research community, giving rise to theoris and models which can claim third-person objectivity, or at least claim that a reasonable disputative process exists to make claims toward objectivity evaluable and, as such, potentially believable. In short, the phenomenologist can claim that an organizing gestalt she thinks is formulative in her experience as both consciously immediate and logicall unified, described from a reflective stance, is not a "subjective" phenomena in the epistemological sense of something idiosyncratic to her, but instead a first-personal phenomenon she can approach as something which others can find also in their own experience; similarly the cognitive linguist can consider her acts of producing or receiving linguistic artifacts, in representative perceptual-situational contexts, bearing in mind that she is a fluent speaker of one or more languages and can anticipate the language-processing practice of others. The very fact of fluency, or competence, implies a certain degree of credibility in ascribing linguistic beliefs to others. Accordingly, so long as we consider a speaker fluent is a language, we commit to finding her basically trustworthy in guessing how fellow speakers would respond to artifacts in that language, and therefore to generalize from subjective judgments outward. This is a good example of how furst-person reflection can be scientificaly operationalized for multi-party disputation; Phenomenology seems to envision a similar dialectic but one expanded outward from language to cognitive acts and experiential groundings in thir totality.

From at least a metascientific perspective, at least, a dialog between Phenomenology and Cognitive Grammar seems promising. Both disciplines topicalize and fine-tune related but complementary methodological issues. Phenomenology, one the one hands, attends more vigorously to a philosophical problemmatics of origination and a "critique of metaphysics", making a willful "bracketing" of extra-experiential assumptions an intrinsic part of its method. Cognitive Grammar may be sympathetic to but not necessarily dependent on similar philosophical performances. For example, an important issue in modern linguistics is whether we have "innate" mental faculties for creating and understanding language or whether language originates as a specialization or repurposing of some other, more general or primordial faculty. Whatever our intuitions about that, surely those are the kind of assumptions that should be "bracketed" in a phenomenology-style analysis. By comparison, while Cognitive Grammar appears to be internally consistent irrespective of any particular opinion vis-à-vis linguistic innateness, I'm not aware of any crucial importance attached to either entertaining or suspending beliefs about innateness (or any other foundational linguistic-cum-philosophical subject matter, like whether a universal grammar underlies individual grammmars or what sort of preceptual inputs qualify as *linguistic* stimuli durin childhood). Certainly Cognitive Grammar methodology is not *constituted* by the *absence* of such belief, the way that Phenomenology is essentially shaped by its counter-traditional, "Cartesian" style.

Notice however that Cognitive Grammar does intrinsically orient analysis to disciplinary commitments — to the accuracy of subjective syntactic/semantics assessments and reports, to the value of hypthetical experiential episodes as proxies for actual ones, to the approximate commonality of experience among a language community — which cannot be just passively accepted (or rejected); they must be actively (even if unconsciously) presupposed in the research process. Phenomenology, given how it thematizes both the legitimacy of (experientially-grounded) presuppositions and the suspension of presuppositions drawn indirectly from scientific or philosophical beliefs, therefore points to a latent contrast in metalinguistic commitments between those that are tangential to and those that are constitutive of the Cognitive Grammar affiliated paradigms: without self-consciously witholding considerations on presumptive "metaphysical" linguistic matters (like innateness vs. linguistic empiricism), Cognitive Grammar does not necessarily to endeavor to classify it's practitioners beliefs on the axis of bracketing vs. disclosure — the commitments that can be suspended partitioned from those that seem either experientially irrefutable or transcendentally prerequisite for any notion of experiential certainty in the first place. Scientists presumably feel that an *entirely* presuppositionless science is impossible, while a more philosophical angle can investigate how, nonetheless, presuppositions vary in their metaphysical posture. Belief in a "poverty of stimulus" suggesting innateness, say — or, conversely, belief that the rich experiential structure surrounding even small-scale language acts presents substantial language-acquisitional stiumli (thereby falsifying the premise of poverty and that avenue of advocacy for innateness) — can both be claimed to stand differently in the ebb and flow of cognitive-linguistic metatheory than commitments to, say, the disputational objectivization of first-person reports.

At the same time, Cognitive Grammar can be claimed to illuminate a wider, or at least a different, intellectual spectrum for the objectivizing dialectic than Phenomenology alone. What I mean is that Phenomenology's path from subjective reflection to rationalistic claims is fairly repetitive: philosophers discuss specific hypothetical/prototypical episodes of consciousness and then write about them in an academic style which invites community reflection and to be read against certain scholarly traditions. Cognitive Grammar does all that too, but there are introduced by virtue of its linguistic subject-matter other discursive facets: the schematic representation of linguistic structure (e.g. parse trees); the diagramming of perceptual-situational gestalts; even technological applications... Scolars can debate which tactics for representing language structure are most suited or most organically consistent for describing language artifacts via the Cognitive-Grammatic lense. The representations specific to Cognitive Grammar can be contrasted with those developed in other grammar theories and their correlative research programs, such as Combinatory, Dependency, and and Head-Driven Phrase-Structural Grammars. These disparate programs are, also, not exclusively focused on explanatory revelations about human language — contemporary linguists also have practical concerns, like text mining and Natural Language Processing. One question for semantic and syntactic theories is whether they are buttressed by computational evidence — whether software created according to the language models and representations are successgul for useful tasks, like classifying documents, machine translation, or pulling information from Natural Language resources.

Cognitive Linguistics in general does not need to mimic these practical concerns — after all, if language understanding does indeed build off of integrated percepual and experiential situatedness, we should be skeptical about the prospects for computers and "Artificially Intelligent" agents to parse language with anything like native fluency (even setting aside how a machine's language processing, correct or not in terms of our desired practical ends, cannot be equated to human empathic understanding). Contrariwise, however, we can also speculate that if the gist of Cognitive Grammar is how a plethora of environing situations — the spatial and force-dynamic arrangements trenchant to our construal of practical scenarios — can be sorted into a relatively small set of canonical schema, corelated with details like noun tense and morphosyntactic modes of agreement, then a computational treatmnt of language need not percive the full spectrum of situational possibilities in full nuance, but only ascertain which one of several schema are evidenced in a given linguistic artifact. Cognitive Grammar would seem to support intuitions that truly humanlike "AI" Natural Language Processing is an unrealistic goal but that, at least potentially, computational NLP can be achieved to some useful approcimation, and employing cognitively realistic languag models can help that project. But I have no firm commitments in either direction; my point is that the technological and computational dimensions of linguistics are relevant for Cognitive Grammar whether or not this approach is sympathetic to the, at times, reductionistic paradigms of conventional formal/computational linguistics. Unlike pre-communicative experience, language is structured by the rigors of coordinated thought and normative signification — as language users we must shoehorn private experience into prototypes that can be expressed to others. Language, in effect, reveals that even rawly immediate experience has a logical order that, to some approximation, can be studied in abstraction — first because language merely by existing shows that this abstracting does happen, in one fashion; and second because language is a platform for studying it. Via language we can empirically study the articulations of conscious experience transcribed into a normative semiosis.

Phenomenology, also, gathers consciousnss as reflected upon into something partially abstract, or with a structure that can be abstractly re-considered. There are some parameters of organization that seem so primeval to experience as to be indubitable. One is the episodic nature of perception and of deliberate action; mental phenomena seem neither sliced into instantaneous moments nor streaming in expansive segments. Instead, conscious reality is a chain of continuous but brief interludes: in one hypothetical kind of scenario, I open a door, which exercises one family of sensate and kinaesthtic attitudes; then I walk across th room; then I look out the window, and so forth. Such episodes stick together in language and maybe in memory, if I have occasion to say, or to later recall, that "I walked into the room" — the state of affairs I commit to memory or verbiage is actually a sequence of smaller episodes. This in turn issubsume into larger units, like when a narration reads "I walked into my living room, sat at my desk, and began to read". What stands out inviolably from this episodic bricolage is — minimally the temporality of, but more consequentially, the multiple scales of the temporality of consciousnss. Episodes link together and sequehtially coalesce into a more durational scale. Phenomenology marks this scale-transition terminology; *protention* and *retention*, for instance, connoting the episodic unity of consciousness over short time spans, is contrasted with concpts like *memory* and *anticipation*.

Meanwhile, an equally salient structuring principle lies in the epistemic attitudes that are carried into perceptual/enactive episodes: consider the contrast between casually looking at the window expecting to see the same scene as any other afyernoon and looking inquisitively because I hear a sound outside. Or maybe I see Hugo sleeping on the sofa, a "seeing" which is actually somewhat inferential, because he's always sleeping on the sofa this time of day. I don't need to confirm — if I briefly glance and see him lying still and relaxed — that the tabby cat on the couch is Hugo and that he's sleping, for me to think to myself (and maybe say to someone else) that "Hugo is sleeping" or "Hugo is sleeping on the sofa". Of course, such a scenario an be altered: if we have two tabby cats then I may want to look more discriminately to ascertain that it is Hugo on the sofa; or I may want to observe him for a few seconds to check *if* he is sleeping, which is a different perceptual event and a different conscious experience that passively seeing "Hugo sleeping" in the throes of an assuming that he is, in fact, sleeping, by analogy to how seeing daylight out the window is *passive* if it's mid-afternoon but may be a premeditatd investigation if I have just woken up and don't know the time.

Here, too, is an apparently "transcendental" or "categorical" (allowing a Kantian echo) pattern in consciousness — the sliding scale of activeness and passiveness in perceiving; the varying degree of epistemic thematization saturating different parts of experience. There are things we take for granted based on the familiarity of our most typical environments, becoming part of the propositional hum of things generally disclosing themslves in ways that carry information but no surprises. There are other things we are consciously aware of wanting to know. Some of our information about the world comes from presumptive expectations that we only really register when they are violated — say I suddenly see Hugo, having awaken, jump down from the sofa; now I have to update my baseline conception of how things are obtaining in my immediate surroundings. In that case I would not be startled to hear him scratching the chair, as compared to if I had not seen him wake up: our construals are woven from baseline assumptiona and then the occasional anomalies where my assumed picture must be revised. But layered within this cycle of passive experience and disruptive revision is another genre of epistemic episodes which is more deliberate and proactive; planning perceprual encounters around specific, epistemically thematized concerns, like looking out the window to trace an unfamiliar sound. This scale of epistemic passivity and activity can be set against — and in consciousness interweaves with — the scales of temporality.

We have then at least this matrix — the double-scale of temporal granularity and epistemic attentiveness — which seems to emanate from the depths of consciousness and yet have something like an abstract structure. Looking over the phenomenological (and philosophy of mind) tradition, we can find other such prereflective abstracta, like the contrast between hyletic sense-data (the purely embodied qualitative character of red or of silky smoothness) and qualia as propositional content (*red* and *smooth* as predicates); or the contrast between fully private experience (like corporeal sensations) and intersubjectively negotiated perception — we may not see the same scenes in exactly the same ways, but at least in productively *similar* ways, if we are in similar locales and vantage-points. A full catalog of these structures would involve a historical overview of Phnomenology as a whole, which is divergent from my aims in this paper. I want to highlight, however, that Phenomenology produces a philosophical narrative about certain ur-structures — described, if all goes well, not from prior scientific or world-view commitments but from commonsensical meditations on (mundane, everyday) consciousness — which can be thematized with the extra performative precision of scholarly discourse; with the care of finding the right language, entrenching usages, diagramming the conceptual relations which should be easier to memoize precisely because of terminological rigor ("protention"; "episodic"; "epistemic attitude"; "enactment"). These structurs get lifted out of the experiential realm and begin to condensate into an abstract regime, something defended like intellectual turf, discussed and analyzed.

Cognitive Grammar also has ur-structures — for sake of argument, consider on one hand the linguistic double-parameters of spatial and force-dynamic schema and, on another, the phenomenological double-scales of temporal granularity and epistemic thematization. Once entered in the intellectual ledger, the cognitive-linguistic schema become subject to a range of formal systemizations, from technical representations of language structures to lexicons, compilations of parts of speech and type-theoretic lexical classifications as well as of inter-word relations, even computer code for representing or automatically building representations of language artifacts. There have not been thus far analogous options for elaborating the phenomenologist's ur-structures — no code-repositories curated by Phenomenology research groups, no technological enhancements of phenomenological literature, for the most part. This is doubtless a manifestation of the historical context where Phenomenology has emerged, but we are now in a different context. Perhaps some 21st-century phenomenologists will engage with computers as agressively as Husserl engaged with mathematics in the *Formal and Transcendental Logic* or the *Investigations*.

Language, in short, is — at least on one level — a *formal system*, notwithstanding that it is also a human medium possessing an arguably irreducible layer of human subtlety and contxt-sensitivity. Linguistics in general, and perhaps Cognitive Grammar in particular, has to bridge experience and formalization. Spatial and force-dynamic schema are — for example and so to speak — one one side of the bridge as perceptual gestalts; but their formal trace is excavated analytically through syntactic and morphosyntactic theories; and a systematic model of morphosyntax involves something like formal language-representation. Insofar as parsing structures are described via "parse graphs" — the parsed sentence notated, with the supplemental data marking the contrast between a sentence spoken in an unstudied human context and the sentence as an object of analysis, as a graph of inter-word relations — then morphosyntactic agreement is a paramter manifest in specific inter-word pairs. So morphosyntactic agreement qua linguistic phenomenon straddles the experiential and formal realms — it can be studied in terms of what perceptual situations call for a particular noun or verb to be aligned with a particular verb, adjectuve, or adverb; it can also be studied as an enrichment of a formal structure. Through such formal strata Cognitive Grammar is not siloed in its study of perceptual groundings, but widens its disciplinary circle to reach a constellation of formal techniques that resonate with grounding experiential intuitions in suggestive ways, revealing both the potential and limits of formalization.

I would like to juxtapose this interdisciplinary continuation of Cognitive Grammar with the academic dissemination of Phenomenology — how the professional philosophical milieu where Phenomenology is mostly practiced shapes its transition from first-person speculation to an intellectual enterprise which, at least in its ways of organizing social resources, operates as a science. Cognitive Grammar operates in a similar milieu, but the strata of formalization evident in linguistics adds a wrinkle that has no direct analog in philosophy. Alongside the discursive and institutional norms which guide the systematization of Cognitive Grammar — the discipline of acdemic writing, peer review, conferences — there are also structural norms sited in linguistic formalizations through which cognitive theories can be explored. This is a development which phenomenologists should observe carefully, because the structural elaboration upon phenomenological research can be augmented beyond *just* inter-textual disputation in an academically curated speech situstion, and can engage with formal systems as logical and technological artifacts. Scholars, of course, debate, refine, and mathematize formal systems in similarly self-conscious academic circles; but insofar as they are modeled and simulated in technological and computational environments, formal systems *also* become technical artifacts which can be explored and manipulated. Alongside the "Communicative Rationality" of academic performance, they buttress a *technical* rationality of implementing computer models, crafting the algebra of formal systems, codifying inter-system translations, and so forth. Computational linguistics, to take one example, is a technical as well as an intellectual practice.

The paradigms of these research programs that emphasize engineering over academic disputation don't necessarily align seamlessly with Cognitive Grammar, and still less with Phenomenology. Despite their *internal* commitment to "bracketing" phenomenologists are still educated participants in the modern world and many presumably do accept in broad outlines the scientist's worldview. Many phenomenologists probably "unofficially" believe that mental phenomena have mundane neurophysical explanations (they are not magic or divine revelations or flux in a sacred ether), however opaque to consciousness itself. So Phenomenology is not *constitutively* antagonistic to a "natural science" of cognitiona and perception. In the Philosophy of Mind, moreover, the path to materialistic perspectives on mental phenomena seems to diverge in two irections — one being the direct study of material systems themselves that we reasonably believe are seats for the neurophysical correlates of consciousness, like neurons and synapses; the other is a mkre functioanlist attemp to describethe systematic organization of such material systems, on the premise that it is easier to make scientific process by seeing the brain (or the entire embodied nervous system) as a large-scale functional system than by reductively studying the biology and physics of microscale constituents, like nerce cells. Sure, we may believe that vision is driven by cells in the eye and optic nerve, as well as specific brain regions; but the explanatory gap between whatever organic properties we may discover researching these bodies and the lived immediacy of visual qualia seems no less expansive. At least as a supplement to such microphysical investigation, functionalist methodologies can potentially narro, even if not elimibate, such explanatory gap. In Cognitive Grammar, for instance, the subtle variation between schematically similar situations is in a sense transcendended by morphosyntactic rules; our language-forming process of mapping perceptual givens to morphosyntactic and lexical prototypes and finding a communicatively stable encoding for them certainly seems amenable to functional description: such meta-cognitive finessing of our preceptual surround is a good candidate of a *functionality* available to our minds, as we are (and in our being) intelligently intersubjective and information-processing life-forms. Insofar as we have formal presentations of perceptual and enactive structures that seem both subjectively realistic *and* faithful to a kind of inner logic, we have the possibility for a rapproachment between Phenomenology and cognitiv functionalism, the two paradigms bridged by the *functional* utility of cycling between immediate and subtly particular experience and formally tractable concept-systems we us to construct productive precis of our environing situations — which in turn can be modeled as formal systems and investigated in that light. Language bears witness to this experiential-to-formal-to-experiential rotation in an especially ubiquitous and well-structrued topos, which is why linguistic methodologies like Cognitive Grammar can be a useful case-study for phenomenological formalization in general.

The preceding discussion has set forth why I believe the formalizing strategies available to Cognitive Grammar should be interesting to Phenomnology, in part because they are perforce available to Phenomnology as well. A formalized cognitive linguistics may not be immediately a formalized Phenomenology, but but it comes tantalizingly close to that, as I hope to show before the end of this paper. But I have until now been addressing this formalization obliquely, talking *about* Phenomenology and Cognitive Grammar as scholarly phnomena rather than *within* either (or both). Here, for most of what follows, I will focus in on one specific formal development of Cognitive Grammar from both a cognitive and computational perspective, which I will then conclude by placing in a more classically phenomenological context.

Part I Cognitive and Computational Process

Any attempt to bridge Computational Linguistics and Cognitive Grammar or Phenomenology must solicit one or several "founding analogies", linking phenomena on the formal/computational side with those on the cognitive/computational side. Here, I will start from the analogy of *cognitive* and *computational process*, or generically "process" (of either variety). Processes, per se, I will leave undefined, although a "computational" process can be considered roughly analogous to a single function implemented in a computer programming language. Th story I want to tell goes something like this: understanding language involves many cognitive processes, many of which are subtly determined by each exact language artifact and the context where it is created. Properly understanding a piece of language depends on correctly weaving together the various processes involved in understanding its component parts, and the structure of the multi-process intergration is suggested by the grammar of the artifact. Grammar, in a nutshell, uses relationships between words to evoke relationships between cognitive processes.

My formal elaboration of this model will be inspired at an elementary level by process *algebra* in the computational setting, but more technically by applied *type theory*. Inter-process relations are the core topic of Process Algebra, including sequentiality (one process followed by another) and concurrency (one process executing alongside another). In practice, detailed research around Process Algebra seems to focus especially on concurrency, perhaps because this is the more complex area of application (designing computer systems which can run multiple threads in parallel). It is likewise tempting to imagine that cognitive-linguistic processes exhibit some degree of parallelism, so that the various pieces of understanding "fall into place" together as we grasp the meaning of a sentence (henceforth using *sentence* as a representative example of a mid-size lanuag artifact in general). Nevertheless, I will focus more on *sequential* relations between processes, suggesting a language model (even if rather idealized) where cognitive processes unfold in a temporal order.

On both the cognitive and computational side, temporality is relative rather than quantified: the significant detail is not "before" and "after" in the sense of measuring time but rather how one process logically precedes another in effects and prerequisites. No theoretical importance is attached to *how long* it takes before processes finish, or how much time elapses between antecedent and subsequent processes (in contrast to subjects like optimization theory, where such details are often significant). We can set aside notions of a temporal continuum where subsequent processes occupy disjoint, extended time-regions; instead, one process follows another if anything affected by the first process reflects this effect at the onset of the second process. Time, in this sense, only exists as manifest in the variations of any state revelant to processes — in the computational context, in the overall state of the computer (and potentially other computers on a network) where a computation is carried out. Two times are different only insofar as the overall state at one time differs from the state at the second time. Time is *discrete* because the relevant states are discrete, and because beneath a certain sclae of time delta there is no possibility of state change.

Analogously, in language, I suggest that we set aside notions of an unfolding process reflecting the temporality of expression. Of course, the fact that parts of a sentence are heard first biases understanding somewhat; and speakers often exploit temporality for rhetorical effect, elonging the pronunciation of words for emphasis, or pausing before words to signal an especially calculated word choice, for example. These data are not irrelvant, but, for core semantic and syntactic analysis, I will nonetheless treat a sentence as an integrated temporal unit, with no value atributed to temporal ordering amongst words except insofar as temporal order establishes word order and word order has grammatical significance in the relevant natural language/dialect.

While antecedent/subsequent inter-process relations are among those formally recognized in Process Algebra, this specific genre of relation is implicit to other models important to computer science, such as Type Theory and Lambda Calculus. If *type-T* is a type, then any computational process which proceduces a value of type *type-T* has a corresponding ("functional") type (for sake of discussion, assume a "value" is anything that can be encoded in a finite sequence of numbers and that "types" are classifications for values that introduce distinctions between functions — e.g., the function to add two integers is different than the function to add two decimals; more rigorous definitions of primordial notions like "type" and "value" are possible but not needed for this paper). Similarly a process which takes as *input* a value of *type-T* is its own type. If two processes have these two types respectively — one outputs *type-T* and the other inputs *type-T* — then the two can be put in sequence, where the output from the antecedent becomes the inut to the subsequent. In this manner inter-process sequential relations become subsumed into "type systems" can can be studied using type-theoretic machinery rather than Process Algebras or Process Calculii as such.

There also exists a robust type-theoretic tradition in (Natural Language) semantics, which is disjoint from but not entirely irrelevant to the type systems of formal and programming languages. Semantic types are recognized at several different levels of classification, but some of the most interesting type-theoretic effects involve medium-grained semantic criteria that are more general than lexical entries but more specific than Parts of Speech. For example, the template *I believed X* generally requires that *X* be a noun (*?I believed run*), but more narrowly a certain *type* of noun, something that can be interpreted as an idea or proposition of some kind (*?I believed flower*). Asher and Pustejovsky point out the anomaly in a sentenc like "Bob's idea weighs five pounds" (ex. 2 p. 5), which possesses a flavor of unacceptability that feels akin to Part of Speech errors but are not in fact syntactic errors. The object of *weigh* is "five pounds" and its subject is "Bob's idea", which is admissible *syntactically* but fails to honor our semantic convention that the verb "to weigh" should be applied to things with physical mass (at least if the direct object denotes a quantity; contrast with *Let's all weigh Bob's idea*, where the *idea* is object rather than subject). These conventions are analogous to Part of Speech rules but more fine-grained: there is a meaning of *weigh* which has (like any transitive verb) to be paired with a subject and object noun, but beyond just being nouns the subject must be a physical body (in effect a sub-type of nouns) and the object a quantitative expression (another sub-type of nouns). Potentially, type restrictions on a coarse scale (e.g. that the subject of a verb must be a noun) and those on a finer scale (as in this sense of *to weigh*) can be unified into an overarching theory, which spans both grammar and semantics — for instance, both Part of Speech rules and usage conventions of the kind often subtly or cleverly subverted in metaphor and idioms (see "flowers want sunshine", "my computer died", "neutrinos are sneaky", as rather elegantly compactified by assigning sentient states to inert things). This is one way of reading the type-theoretic semantic project.

Along with Process Algebra, my take on linguistic understanding is informed by type theory (in both computational and linguistic contexts), but particularly by the merged notion of *typed* processes. So if we say that something has the *type* of a physical-body noun — that "Physical Body" is a type in the overall semantics of language — then I propose a corresponding type of cognitive (perceptual and conceptual) processes characteristic of perceiving and reasoning about physical things. A particular designatum — a bag of rice, say — is subsumed under the semantic type insofar as our perceptual encounters with that thing — and/or our concepual exercises pertaining to its properties and proclivities (like being difficult to carry) — are roughly prototyped by a certain generic kind of cognitive process. This assumes that there is a similitude among processes of perceiving and thinking about physical bodies (at least the mid-sized, quotidian physical things that tend to enter nonspecialist language) sufficient to subsume them under a common prototype, which I then argue forms the cognitive substratum for the semantic type "Physical Object". Moreover, I contend a similar cognitive substratum can be found for other mid-scale semantic types that underlie analyses of semantic acceptability and metaphoricality, like "Living Thing", "Sentient Living Thing" ("flowers want sunshine" is metaphorical because it ascribes propositional attitudes to something whose type does not literally support them), and "Social Institutions" ("The newspaper you're reading fired its editor" exhibits a "type coercion" where *newspaper* is read first as an object and then as a company). One feature of semantic types is the lexical superposition of different types to produce what (in a slightly different context) Gilles Fauconnier calls a "blend": in "Liverpool, which is near the ocean, built new docks", the city is treated as both a geographic region and a body politic.

"Weighs", too, as a verb, can be given a typed-process interpretation. In its least metaphoric sense, "to weigh" connotes a practical action of measuring some object's weight by using something like a scale; as *cognitive* process the verb embodies are ability to plan, reflect upon, or contemplate this practice. So an "idea weighing 5 pounds" is anomalous because it is hard to play out in our minds a form of this practical act where the thing being weighed is mental. Howevere, there are plenty of more figurative uses related to "weighing ideas", "heavy ideas", and so forth, so we are able to isolate the dimension of "judging" and "measuring" which is explicit in literal "weighing", and abstracting from the physical details use "weigh" to mean "measure" or "assess" in general. The phrase "weigh an idea" therefore connotes its own cognitive process — imagining someone thinking about the idea in an evaluative way — but this figurative "script" is closed off by "5 pounds" which forces us to conceive the weighing literally with a scale, not figuratively as a kind of mental assessment. Once again, the type anomaly can be seen as a failure to map the linguistic senses evident in a sentence to an internally consistent set of cognitive procedures for dilating the semantic content seeded within each word.

Notice that I am treating cognitive processes, in themselves, as semantic more than grammatical phenomena. Literally, weighing something is a multi-stp act (lifting it on the sale, reading the measurement), and even in our mental replay of hypothetical weighing-acts it seems impossible not to imagine distinct phases (just as it is impossible not to picture left and right sides of an imaginary cow). Howevere, I assume that the cognitive script is figured by the lexeme "weighs" as a connotative unit: whatever internal structure our mental script of "weighing something" has, this structure is not a *linguistic* structur that must be encoded grammatically. Similar, the concept *buttered toast* suggests a confluence of perceptual, physical-operational, and concptual aspects — we are incliјed to regard toast as *buttered* if it looks a certain way and also if we have seen someone apply butter to it (or have done so ourselves) and also if we are in a context where we expect to find toast that may be buttered (we are not disposed to call a piece of bread in a grocery store "buttered toast" even if it has that appearence). So the adjective "buttered" introduces multiple cross-modal parameters in addition to the underlying concept "toast"; but I feel that the lexeme aggregates these parameters into a single *linguistic* unit. In Langacker's terms, the various elements of "buttered" do not suggest *constructive effort*, as if deliberate *linguistic* processing were needed to unpack the linguistic entity to its constituent parts. Instead, "buttered" functions *semantically* as a unit (and likewise syntactically as the unit entering relations with other words — e.g. buttered-toast is an adjective/noun pair, not the noun "butter" at the root of the adjective) — even if its cognitive process is multi-faceted. Indeed, this is precisely the signifying economy of language: it captures complex cognitive procedures by iconic, repeatable lexical units.

On that theory, tieing specific word-senses to stereotyped cognitiv processes is a matter of semantics, not grammar per se. Grammar, I contend, come into play when multiple processes need to be integrated. The concept "buttered toast", for example, seems to start from a more generic concept (toast) and then add extra detail (the buttering, with all that implies conceptually, pragmatically, and sensorially). This is suggested by the substitutability of just "toast" for "buttered toast":

▾ (1)I snacked on toast and coffee.

▾ (2)I snacked on buttered toast and iced coffee/.

Because the first sentence is perfectly clear, it seems that the ideas expressed (at least in this context) by *toast* and *coffee* are reasonably complete in themselves, so the adjectives have the effect of starting with a complete idea and adding on etra detail. Procedurally, then, it seems like we have some process which takes us to "toast" and "coffee" and then, subsequent to that (logically if not temporally) we add the wrinkle of re-conceivingbthe toast as buttered and the coffee as iced. In short, the adjective-noun pairing is compelling us to run a pair of cognitive proceesses in sequence, one establishing the noun-concept as a baseline and one adding descriptive detail by an "adjectival", a specificational process.

Counter to that analysis, someone might judge that phrases like "buttered toast" and "iced coffee" are conventional enough that we don't interpret them through two meaningfully disjoint processes. This is entirely possible, given how erstwhile aggregate expressions bcome established units — what Langacker calls *entrenchment*. Different phrases ehibit different levels of entrenchment:

▾ (3)I snacked on toast and instant coffee.

▾ (4)I snacked on toast and Eritrean coffee.

Arguably "instant coffee" is a de facto lexical unit, partly because reading it in terms of constituent parts is rather nonsensical (there's no non-oblique way to understand "coffee" being qualified as "instant"). Surely, however, "Eritrean coffee" is heard as a compound phrase (at least in 2019 — it is unlikely, but not impossible, that future Eritrean coffee growers will be so successful that we hear the phrase as a brand name or culinary term of art, like "Hershey's kisses" or "French toast"). The status of "iced coffee" is probably somewhere between those to. But to the degree that a language element (whether word oe phrase) is entrenched and generally processed linguistically as a unit, I maintain, it tends to be governed by an integrally complete cognitivee process — not necessarily one without inner structure, but where the elements of this structure piece togther perceptually and situationally, rather than seeming to be *linguistically* disjoint conceptualizations that are brought together by the shape of linguistic phrases. Conversely, where a cognitive process has this integral character, discursive pressures nudge the language toward entrenching some descriptive phrase as a quasi-lexeme; what starts being heard as a compound designation evolves to the point where language users don't attend to constituent parts.

Obviously, this theory presupposes that there is an available distinction to be drawn between a "procedural" synthsis of disparate cognitive processes and a perceotual and/or conceptual synthesis constitutive of individual cognitive episodes. Phenomonology seems to back this up — there are some conceptual compounds that seem more episodically fused than others. Buttered toast may evoke a temporally not-quite-instantaneous conceptualization — at the core of the concept is a practical activity that takes a few seconds to complete — but we also can imagine the buttering-act apprehended in one sole episode. On the other hand, "Eritrean coffee" ties together concepts of much more scattered provnance; the perceptual unity of *coffee* (in the sense of a spcific liquid in a specific container) along with the abdtract geopolitical "background knowledge" implicit in the adjective "Eritrean". As a cognitive synthesis "Eritrean coffee" is conceptual rather than perceptual. Provisionally we can treat this in the context of "buttered toast" being a partially-entrenched phraseology while "Eritrean coffee" is undeniably a phrasal compound, something whose constructive form must beparsed linguistically rather than figuratively.

This analysis, though, needs many cavats. After all, many bonafide *phrases* (not "quasi-lexemes") nevertheless exhibit significant Phenomenological unity — i.e., they evoke integral perceptual complexes: *big dog*; *hot coffee*; *speeding car*; *red foliage*. Linguistically these seem like an underlying concept acquiring perceptual specificity via adjectival modification: "hot" was how the coffee came to my exprience because I experienced it as hot (it wasn't like I experienced the coffee and then had to contemplate whether it is hot or cold). After all, coffee must be experienced as hot, cold, or lukewarm; it cannot be eprienced without temperature (assuming I am coming into contact with it and not just looking at it). Similarly a car must be sen as at rest, moving slowly, or speeding; foliage must be seen as having some color. I have argued, however, that unless entrenched as idiomatic phrases adjective-noun pairs like *hot coffee* or *buttered toast* should be read as grammatical complxes and accordingly (in my thoey) as junctures between distinct cognitive processes. On the other hand, I argued that "instant coffee" was effectively entrenched *because* there is no simplistic concepual unity between "instant" and "coffee", which makes it harder to hear the phrase as descriptive. Instead, the semantics of that particular adjective-noun connection are circuitous and a little hyperbolic: "instant" coffee is coffee as a substance (not a drink, in that state) from which coffee the drink can be quickly (but not instantaneously) prepared. There is a lot going on the semingly simple "instant coffee": the shift from coffee-as-substance to coffee-as-drink; the "instant" exaggeration. In this case, the adjectival modification has *so many* moving parts that, I'm inclined to argue, it is hard to cover the whole scenario with a dscriptive phrase; which in turn creates selective pressures for some pseudo-lexical unit to emerge (which turned out to be "instant coffee") as a mnemonic for the whole conceptual multiplex. Indeed conceptually intricate wholes tend to quickly acquire pithy conventional nominalizations simply for rhetorical convenience ("Brexit Negotiations"; "Quantum Gravity"; "International Transfer Window"; "#metoo").

Notwithstanding these variations, I still find a certain logic in the relation between phenomenological unity and semantic entrenchment. Perceptually integrated wholes may correlate with linguistically aggregate constructions insofar as there is a transparent logical destructuring in the perceptual unity: in the case of substance-attribute pairs (like "hot coffee") — deferring in the phenomenological context to Husserl's account of dependent moments — there is a basically unsubtle distinction between an underlying concept (like coffee) and the qualities which are its mode of appearance as well as conceptual predicates (like hot, cold, black, or light, describing sensory properties innate to the experience of a coffee-token as well as state-reports that can be propositionally attributed to it). Although the minimal sensate intention of the coffee and the predicative disposition toward ascriptions like *black* and *hot* are consciously intertwined, surely I am aware of a logicality in experience that gives the sensate and predicative dimensions different epistemic status. I don't think of my experience of the coffee's being hot as just a hot sensation qua medium of my sensorily apprehending the coffee, but rather as the sensate mechanism by which I observe the apparent fact that the coffee is hot, as a state of affairs and not just as subjective impression. We are constantly extrapolating our perceptual encounters to propositional content along these lines. As such, I contend such an (in some sense) innate perception-to-predication instinct grounds the procedural slicing of linguistic processes: *hot coffee* retraces ina linguistic construction the logical order of a coffee perception which on one level is a raw perceptual encounter but is simultaneously a predicative attribution. "Hot coffee" denotes a substancs that can be experienced in the mode of a base concept (coffee) which is given predicative qualification (the coffee *is* hot). The fact that there may not be a noticed temporal gap in *experience* between the sensate perception and the epistemic posture does not preclude a certain logical antecedent-subsequent ordering: the concept "coffee" is the predicative base for my propositional attitude that what I am dealing with here is hot coffee, not hot-sensations-disclosing-coffee or coffee-I-experience-as-hot (but who knows, maybe I'm hallucinating) or any other artificial skeptifying of my actual experience, which is of raw perception pregnant with propositional content.

So I wishbto justify claims that (non-entrenched) phrases complexes like "hot coffee" are unions of disjoint cognitive processes by noting that while such phrases evoke a certain perceptual unity, they evoke a *kind* of unity which we habitually regard *conceptually* as divided into sensate givenness founding epistemic attitudes. Cognitive processes are not exclusively perceptual; they are some mixture of perceptual and conceptual (and enactive/kinaesthetic/operational). A perceptual unity can cover two conceptual aspects, like a sheet covering two mattresses. So the perceptual unity of hot coffee can become the conceptual two-step of coffee as substance and hot as attribute; committing this unity to cognition as an overarching lifelong faculty involves registering a thought-process of coffee as a substance which can, in acts of logical predication, be believed to be hot or cold, black or light, etc. The apprehension of the substance is a different cognitive process than the predication of the attribute, in terms of how these mental acts fit within our epistemic models, even if these two processes are experientially fused. Typically we see the coffee before we touch or taste it, so already the coffee has a logical status apart from the heat we predicate in it. Likewise, even if the black color is inxtricable from our perceiving (apart from odd situations where we drink the coffee without looking at it), we know the color will change if we add milk (even if just in principle because, preferring black coffee, we don't actually do so); so we know the coffee has a logical substrate apart from its color too. All of this ideation is latent in the coffee-perceptions notwithstanding whatever perceptual unities we experience that cloak logical forms like substance/attribute under the inexorable togetherness of disclosure (the phenomenological impossibility of spatial expanse without color, say). In short, disjoint cognitive processes can be required to reconstruct a perceptually unified situation or episode, insofar as wec are not just living through the episode but prototyping, logically reconstructing, signifying it — the perceptual unity in the moment does not propagate to procedural atomicity in absrbing the episode into rational exercises.

Experience, then prsents *both* perceptual unities and cognitive-propositional multiplicity; language can inherit both holism on the perceptual side and compositionality on the rational side, even in a single enactive/perceptual episode. Depending on how we via language want to figure and express experience, we can bring either unity or compositionality to the fore. Our linguistic choices will evoke perceptual unity if they select entrenched word-senses or quasi-lexical forms; they will evoke compositionality if they gravitate toward compound phrases and complex, relatively rare lexicalizations and modes of expression. To the degree that we are interested in a cognitive-phenomenological *semantics*, we can attend to the first part of this equation, to how the understood atomicity of a word sense or a conventionalized phrase often suggests an object or phenomenon consciously apprehended as an integral whole; we can trace phenomenologically the apperceptive unity that seems to dribve the linguistic community's accepting lexical atoms in this sense. Conversely, to the degree that we are interested in a cognitive-phenomenological *grammar*, we can attend to how logically composite predication emerges even within perceptual unity, because our encounter with phenomena is not (save for exotic artistic or meditative pursuits) the "dasein" of irreflective sensory beings immersed in a world of pure experience but the deliberate action of epistemic beings carrying (modifiable but not random) propositional attitudes to perceptual encounters.

Modeling grammar as a coordination between cognitive processes may be an idealization, precisely becaue the compositive and integrative faces of consciousness are two sids of the same coin: it's not as if we work through a thought of "coffee" or "toast", abstract and without sensory specificity, noticeably prelude to conceived/perceived attributes like "hot", "cold", or "buttered". But we can still ascribe to linguistic-understanding processes an idealized, "as if" temporality, treating the elucidating a sentence as a sequence of procedures leading from bare concepts to well-rendered logical tableau, suffusee with some level of descriptive and situational particularity. So we go from *coffee* to *iced coffee* to *butterred toast and iced coffee* to *snacking on butterred toast and iced coffee*; each link in the chain stepping up toward propositional totality. My point is not that the logical form of the sentence is composed from logically primitive and abstract parts, which is fairly trite; my point is that such logical composition is only apparent after a pattern of cognitive integration that is more subtle and exceptional. Extra-mentally, buttered toast is just toast with butter on it, a fairly simplistic logical conjunction. Read as a baton passed between two acts of mind, however — conciving toast and then conciving it buttered — the conjunction is more elaborate; the cognitive resources of "buttered" are not just "something with butter on it" but the implication of a sensory summation (the flavor, color, scent) and operational narrative (we have seen or performed the deliberate act of applying the butter). Similarly a person dressed up is not just someone whose torso is encircled by articles of clothing; a barking dog is not just an animal making random noises; a stray cat is different from a lost cat. In their interpenetration, cognitive processes develop (in the photographer's sense) narrative and causative threads that are latent in worldly situations but reduced out of logical glosses; that is why it seems incomplete, lacking nuance, or beside the point to explicate semantic meanings in logical terms, like "bachelor" as "unmarried man" (we can certainly imagine a sentence like "My best friend has been married for years but he's still a bachlor", meaning he still has the habits and attitudes of his single days).

A theory of sentences building from conceptual underspecification to logical concreteness does not preclude there being different scales of specificity. "I snacked on toast and coffee" is just as acceptable as "I snacked on buttered toast and iced coffee". The communication conveys as much situational detail as warranted in the conversational, pragmatic context. Language always has the *capability* to push further and further into specificty; how exhaustively the language user avails of this capability is a matter of choice. As theorists of language we must then analyze how language possesses the *latent* capacity to draw ever finer pictures; the adjectival *buttered* toast and *iced* coffee takes the granularity of signifying at one level (the level of the "I snacked on toast and coffee" sentence) and layers on (or really layers *within*) a yet more specific level. The architecture how this happens is well addressed by type-theoretic mehods (both coaese and mid-grained).

Note that "buttered" in "buttered toast" is in the current examples not absolutely essential; the sentences without this qualification are similar in meaning to the sentenes with it. This suggests that if the adjectives designate (or solicit) cognitive processes that build upon some other (logically) prior process, the anterior process can also suffice on its own, at least in the context of these example sentences.

Part II Interpretive Processes and Triggers

My central thesis in this paper is that language understanding involves integrating diverse "cognitive procedures", each associated with specific words, word morphologies (plural forms, verb tense, etc) and sometimes phrases. This perspective contrasts with and adds nuance to a more "logical" or "truth-theoretic" paradigm which tends to interpret semantic phenomena via formal logic — for example, singular/plural in Natural Language as a basically straightforward translation of the individual/set distinction in formal logic. Such formal intuitions are limited in the sense that (to continue this example) the conceptual mapping from single to plural can reflect a wide range of residual details beyond just quantity and multitudes. Compare *I sampled some chocolates* (where the count-plural suggests *pieces* of chocolate) and *I sampled some coffees* (where the count-plural implies distinguishing coffees by virtue of grind, roast, and other differences in preparation) (note that both are contrasted to mass-plural forms like *I sampled some coffee* where plural agreement points toward material continuity; there is no discrete unit of coffee qua liquid). Or compare *People love rescued dogs* with *People fed the rescued dogs* — the second, but not the first, points toward an interpretation that certain *specific* people fed the dogs (and they did so *before* the dogs were rescued).

The assumption that logical modeling can capture all the pertinent facets of Natural-Language meaning can lead us to miss the amount of situational reasoning requisite for commonplace understanding. In *People fed the rescued dogs* there is an exception to the usual pattern of how tense and adjectival modification interact: in *He has dated several divorced millionaires* it is implied that the ladies or gentlemen in question were divorced and millionaires *when* he dated them: that the events which gave them these properties occurred *before* the time frame implied (by tense) as the time of reference for the states of affairs discussed in the sentence (jokes or rhetorical flourishes can toy with thse expectations, but that's why they *are*, say, jokes; consider the dialog: "He likes to date divorced women — I thought they were all married? — Not after he dated them!"). But we read "people fed" in *People fed the rescued dogs* as occurring before the rescue; because we assume that *after* being rescued the dogs would be fed by veterinarians and other professionals (who would probably not be designated with the generic "people"), and also we assume the feeding helped the dogs survive. We also hear the verb as describing a recurring event; compare with *I fed the dog a cheeseburger*.

To be sure, there are patterns and templates governing scope/quantity/tense interactions that help us build logical models of situations described in language. Thus *I fed the dogs a cheeseburger* can be read such that there are multiple cheeseburgers — each dog gets one — notwithstanding the singular form on "a cheeseburger": the plural "dogs" creates a scope that can elevate the singular "cheeseburger" to an implied plural; the discourse creates multiple reference frames each with one cheeseburger. Likewise this morphosyntax is quite correct: *All the rescued dogs are taken to an experienced vet; in fact, they all came from the same veterinary college*: the singular "vet" is properly alligned with the plural "they" because of the scope-binding (from a syntactic perspective) and space-building (from a semantic perspective) effects of the "dogs" plural. Or, in the case of *I fed the dog a cheeseburger every day* there is an implicit plural because "every day" builds multiple spaces: we can refer via the spaces collectively using a plural (*I fed the dog a cheeseburger every day — I made them at home with vegan cheese*) or refer within one space more narrowly, switching to the singular (*Except Tuesday, when I made it out of ground turkey and swiss*).

Layers of scope, tense, and adjectives interact in comples ways that leave room for common ambiguities: *All the rescued dogs are [/were] taken to an experienced [/specialist] vet* is consistent with a reading wherein there is exactly one vet, and she has or had treated every dog, as well as where there are multiple vets and each dog is or was treated by one or another. Resolving such ambiguities tends to call for situational reasoning and a "feel" for situations, rather than brute-force logic. If a large dog shelter describes their operational procedures over many years, we might assume there are multiple vets they work or worked with. If instead the conversation centers on one specific rescue we'd be inclined to imagine just one vet. Lexical and tense variation also guides these impressions: the past-tense form ("...the rescued dogs were taken...") nudges us toward assuming the discourse references one rescue (though it could also be a past-tense retrospective of general operations). Qualifying the vet as *specialist* rather than the vaguer *experienced* also nudges us toward a singular interpretation.

What I am calling a "nudge", however, is based on situational models and arguably flows from a conceptual stratum outside of both semantics and grammar proper, maybe even prelinguistic. There appears to be no explicit principle either in the semantics of the lexeme "to feed", or in the relevant tense agreements, stipulating that the feeding in *People fed the rescued dogs* was prior to the rescue (or conversely that *Vets examined the rescued dogs* describes events *after* the rescue). Instead, we interpret the discourse through a narrative framework that fills in details not provided by the language artifacts explicitly (that abandoned dogs are likely to be hungry; that veterinarians treat dogs in clinics, which dogs have to be physically brought to). For a similar case-study, consider the sentences:

▾ (5)Every singer performed two songs.

▾ (6)Everyone performed two songs.

▾ (7)Everyone sang along to two songs.

▾ (8)Everyone in the audience sang along to two songs.

The last of these examples strongly suggests that of potentially many songs in a concert, exactly two of them were popular and singable for the audience. The first sentence, contrariwise, fairly strongly implies that there were multiple pairs of songs, each pair performed by a different singer. The middle two sentences imply either the first or last reading, respectively (depending on how we interpret "everyone"). Technically, the first two sentences imply a multi-space reading and the latter two a single-space reading. But the driving force behind these implications are the pragmatics of "perform" versus "sing along": the latter verb is bound more tightly to its subject, so we hear it less likely that *many* singers are performing *one* song pair, or conversely that every audience member *sings along* to one song pair, but each chooses a *different* song pair.

The competing interpretations for *perform* compared to *sing along*, and *feed* compared to *treat*, are grounded in lexical differences between the verbs, but I contend the contrasts are not laid out in lexical specifications for any of the words, at least so that the implied readings follow just mechanically, or on logical considerations alone. After all, in more exotic but not implausible scenarios the readings would be reversed:

- (9)The rescued dogs had been treated by vets in the past (but were subsequently abandoned by their owners).
- (10)Every singer performed (the last) two songs (for the grand finale).
- (11)Everyone in the audience sang along to two songs (they were randomnly handed lyrics to different songs when they came in, and we asked them to join in when the song being performed onstage matched the lyrics they had in hand).

In short, it's not as if dictionary entries would specify that "to feed" applies to rescued dogs before they are rescued, and so forth; these interpretations are driven by narrative construals narrowly specific to given expressions. The appraisals would be very different for other uses of the verbs in (lexically) similar (but situationally different) cases: to "treat" a wound or a sickness, to "perform" a gesture or a play. We construct an interpretive scaffolding for resolving issues like scope-binding and space-building based on fine-tuned narrative construals that can vary alot even across small word-sense variance: As we follow along with these sentences, we have to build a narrative and situational picture which matches the speaker's intent, sufficiently well.

And that requires prelinguistic background knowledge which is leveraged and activated (but not mechanically or logically constructed) by lexical, semantic, or grammatical rules and forms: *rescued dogs* all alone constructs a fairly detailed mental picture where we can fill in many details by default, unless something in the discourse tells otherwise (we can assume such dogs are in need of food, medical care, shelter, etc., or they would not be describd as "rescued"). Likewise "sing along" carries a rich mental picture of a performer and an audience and how they interact, one which we understand based on having attended concerts rather than by any rule governing "along" as a modifier to "sing" — compare the effects of "along" in "walk along", "ride along", "play along", "go along". Merely by understanding how "along" modifies "walk", say (which is basically straightforward; to "walk along" is basically to "walk alongside") we would not automatically generalize to more idiomatic and metaphorical uses like "sing along" or "play along" (as in *I was skeptical but I played along (so as not to start an argument)*).

We have access to a robust collection of "mental scripts" which represent hypothetical scenarios and social milieus where language plays out. Language can activate various such "scripts" (and semantic as well as grammatical formations try to ensure that the "right" scripts are selected). Nonetheless, we can argue that the conceptual and cognitive substance of the scripts comes not from language per se but from our overall social and cultural lives. We are disposed to make linguistic inferences — like the timeframes implied by *fed the rescued dogs* or the scopes implied by *sang along to two songs* — because of our enculturated familiarity with events like dog rescues (and dog rescue organizations) and concerts (plus places like concert halls). These concepts are not produced by the English language, or even by any dialect thereof (a fluent English speaker from a different cultural background would not necessarily make the same inferences — and even if we restrict attention to, say, American speakers, the commonality of disposition reflects a commonality of the relevant cultural anchors — like dog rescues, and concerts — rather than any homogenizing effects of an "American" dialect). For these reasons, I believe that trying to account for situational particulars via formal language models alone is a dead end. This does not mean that formal language models are unimportant, only that we need to picture them resting on a fairly detailed prelinguistic world-disclosure.

There are interesting parallels in this thesis to the role of phenomenological analysis, and the direct thematization of issues like attention and intentionality: analyses which are truly "to the things themselves" should take for granted the extensive subconscious reasoning that undergirds what we consciously thematize and would be aware of, in terms of what we deliberately focus on and are conscious of believing (or not knowing), for a first-personal exposé. Phenomenological analysis should not consider itself as thematizing every small quale, every little patch of color or haptic/kinasthetic sensation which by some subconscious process feeds into the logical picture of our surroundings that props up our conscious perception. Analogously, linguistic analysis should not thematize every conceptual and inferential judgment that guides us when forming the mental, situational pictures we then consult to set the groundwork for linguistic understanding proper.

These comments apply to both concptual "background knowledge" and to situational particulars of which we are cognizant in rference to our immediate surroundings and actions. This is the perceptual and operational surrounding that gets linguistically embodied in deictic reference and other contextual "groundings". Our situational awareness therefore has both a conceptual aspect — while attending a concert, or dining at a restaurant, say, we exercise cultural background knowledge to interpret and participate in social events — and also our phenomenological construal of our locales, our immediate spatial and physical surroundings. Phenomenological philosophers have explored in detail how these two facets of situationality interconnect (David Woodruff Smith and Ronald McIntyre in *Husserl and Intentionality: A Study of Mind, Meaning, and Language*, for instance). Cognitive Linguistics covers similar territory; the "cognitive" in Cognitive Semantics and Cognitive Grammar generally tends to thematize the conception/perception interface and how both aspects are merged in situational understanding and situationally grounded linguistic activity (certainly more than anything involving Artifcial Intelligence or Computational Models of Mind as are connoted by terms like "Cognitive Computing"). Phenomenological and Cognitive Linguistic analyses of situationality and perceptual/conceptual cognition (cognition as the mental synthesis of preception an conceptualization) can certainly enhance and reinforce each other.

But in addition, both point to a cognitive and situational substratum underpinning both first-person awareness and linguistic formalization proper — in other words, they point to the thematic limits of Phenomenology and Cognitive Grammar and the analytic boundary where they give way to an overarching Cognitive Science. In the case of Phnomenology, there are cognitive structures that suffuse consciousness without being directly objects of attention or intention(ality), just as sensate hyletic experience is part our consciousness but not, as explicit content, something we in the general case are conscious *of*. Analogously, conceptual and situational models permeate our interpretations of linguistic forms, but are not presented explicitly *through* these forms: instead, they are solicited obliquely and particularly.

What Phenomnology *should* explicate is not background situational cognition but how attention, sensate awareness, and intentionality structure our orientation *visavis* this background: how variations in focus and affective intensity play strategic roles in our engaged interactions with the world around us. Awareness is a scale, and the more conscious we are of a sense-quality, an attentional focus, or an epistemic attitude, reflects our estimation of the importance of that explicit content compared to a muted experintial background. Hence when we describe consciousness as a stream of *intentional* relations we mean not that the intended noemata (whether perceived objects or abstract thoughts) are sole objcts of consciousness (even in the moment) but are that within conscious totality which we are most aware of, and our choice to direct attention here and there reflcts our intelligent, proactive interacting with the life-world. Situational cognition forms the background, and Phenomenology addresses the structure of intentional and attentional modulations constituting the conscious foreground.

Analogously, the proper role for linguistic analysis is to represent how multiple layers or strands of prelinguistic understanding, or "scripts", or "mental spaces", are woven together by the compositional structures of language. For instance, *The rescued dogs were treated by an exprienced vet* integrates two significantly different narrative frames (and space-constructions, and so forth): the frame implied by "rescued dogs" is distinct from that implied by "treated by a veterinarian". Note that both spaces are available for follow-up conversation:

▾ (12)The rescued dogs were treated by an experienced vet. One needed surgery and one got a blood transfusion. We went there yesterday and both looked much better.

▾ (13)The rescued dogs were treated by an experienced vet. One had been struck by a car and needed surgery on his leg. We went there yesterday and saw debris from another car crash — it's a dangerous stretch of highway.

In the first sentence "there" designates the veterinary clinic, while in the second it designates the rescue site. Both of these locales are involved in the original sentence (as locations and also "spaces" with their own environments and configurations: consider these final three examples).

▾ (14)The rescued dogs were treated by an experienced vet. We saw a lot of other dogs getting medical attention.

▾ (15)The rescued dogs were treated by an experienced vet. It looked very modern, like a human hospital.

▾ (16)The rescued dogs were treated by an experienced vet. We looked around and realized how dangerous that road is — for humans as well as dogs.

What these double space-constructions reveal is that accurate language understanding does not only require the proper activated "scripts" accompanying words and phrases, like "rescued dogs" and "treated by a vet". It also requires the correct integration of each script, or each mental space, tieing them togther in accord with speaker intent. So in the current example we should read that the dogs *could* be taken to the vet *because* they were rescued, and *needed* to be taken to the vet *because* they needed to be rescued. Language structures guide us toward how we should tie the mental spaces, and the language segments where they are constructed, together: the phrase "*rescued* dogs" becomes the subject of the passive-voice *were treated by a vet* causing the two narrative strands of the sentence to encounter one another, creating a hybrid space (or perhaps more accurately a patterning between two spaces with a particular temporal and causal sequencing; a hybrid narration bridging the spaces). It is of course this hybrid space, this narrative recount, which the speaker intends via the sentence. This idea is what the sentence is crafted to convey — not just that the dogs were rescued, or that they were taken to a vet, but that a causal and narrative thread links the two events.

I maintain, therefore, that the analyses which are proper to linguistics — highlighting what linguistic reasoning contributes above and beyond background knowledge and situational cognition — should focus on the *intgration* of multiple mental "scripts", each triggered by different parts and properties of the linguistic artifact. The *triggers* themselves can be individual words, but also morphological details (like plurals or tense marking) and morphological agreement. On this theory, analysis has two distinct areas of concerns: identification of grammatical, lexical, and morphosyntactic features which trigger (assumedly prelinguistic) interpretive scripts, and reconstructing how these scripts interoprate (and how language structure determines such integration).

In the case of isolating triggers, a wide range of linguistic features can trigger interpretive reasoning — including base lexical choice; word-senses carry prototypical narrativ and situational templates that guide interpretation of how the word is used in any given context. "Rescued", for example, brings on board a network of likely externalitis: that there are rescuers, typically understood to be benevolent and intending to protect the rescuees from harm; that the rescuees are in danger prior to the rescu but safe afterward; that they need the rescuers and could not have reachd safety themselves. Anyone using the word "rescue" anticipats that their addressees will reason through some such interprtiv frame, so the speaker's role is to fill in the details descriptively or deictically: who are the rescues and why they are in danger, who are the rescurs and why they are benevolnt and able to protect the rscuees, etc. The claim that the word "rescue", by virtue of its lexical properties, triggers an interprtive "script", is a proposal that when trying to faithfully reconstruct speaker intentions we will try to match the interpretive frame to the current situation.

The "script" triggered by word-choice is not just an interpretive frame in the abstract but the interpretive *process* that matchs the frame to the situation. This process can be exploited for metaphorical and figurative effect, broadening the semantic scope of the underlying lexeme. In the case of "rescue" we have less literal and more humorous or idiomatic examples like:

- (17)I'm going to rescue her from that boring conversation.
- (18)The trade rescued a star athlete from a losing team.
- (19)Your invitation rescued me from studying all night.
- (20)New mathematical models rescued her original research from obscurity.
- (21)Discovery of nearby earth-like planets rescued that star from its reputation as ordinary and boring and revealed that its solar system may actually be extraordinary.
- (22)His latest comments rescued him from the perception that he never says anything controversial.
- (23)The Soviets rescued thousands of people from a basically-defeated Germany and sent them to Siberia.

Each of these cases subverts the conventional "rescue" script by varying some of the prototypical frame details: maybe the "danger" faced by the rescuee is actually trivial (as in the first three), or the rescuee is not a living thing whose state we'd normally qualify in trms of "danger" or "safety", or by overturning the benevolence we typically attribute to rescue events. In the penultimate sentence someone is described as rescuing *themselves*, but the effect is ironic: he actually caused trouble for himself, or so the speaker clearly believes. And in the final sentence the speaker clearly believes the "rescue" was not needed, that it was not benevolent, and that the "rescuees" ended up worse off: so the choice of word "rescue" is clearly both ironic/sarcastic and implies a mockery of attempts to portray the rescuers as benevolent.

But in these uses subverting the familiar script does not weaken the lexical merit of the word choice; instead, the interpretive act of matching the conventional "rescue" script to the matter at hand reveals details and opinions that the speaker wishes to convey. The first sentence, for instance, uses "rescue" to connote that being stuck in a boring conversation (and being too polite to drift away with no excuse) is an unpleasant (even if not life-threatening) circumstance. So one part of the frame (that the rescuee needs outside intervention) holds while the other (that the rescue is in danger) comes across as excessive but (by this very hyperbole) communicating speaker sentiment. By both invoking the "rescue" script and exploiting mismactches between its template case and the current context, the speaker conveys both situational facts and personal opinions quite economically. Similarly, "rescue a paper from obscurity" is an economical way of saying that research work has been rediscovered in light of new science; "rescued from a bad team" is an economical way of connoting how an athletic career is less fulfilling on a bad team, and so forth.

All of these interprtive effects — both conventional and unconventional usages — stem from the interpretive scripts bound to words (and triggered by word-choice) at the underlying lexical level — we can assess these by reference to lexical details alone, setting aside syntactic and morphological qualities. Of course, then, a host of further effects are bound to morphological details when they *are* considered. Case in point are plurals: for each plural usage we have a concptual transformation of an underlying singular to a collective, but how that collective is pictured varies in context. One dimension of this variation lies with mass/count: the mass-plural "coffee" (as in "some coffee") figures the plurality of coffee (as liquid, or maybe coffee grounds/beans) in spatial and/or physical/dynamic terms. So we have:

- (24)There's some coffee on your shirt.
- (25)There's coffee all over the table.
- (26)She poured coffee from an ornate beaker.
- (27)There's too much coffee in the grinder.
- (28)There's a lot of coffee left in the pot.
- (29)There's a lot of coffee left in the pot — should I pour it out?

These sentences use phrases associated with plurality ("all over", "a lot", "too much") but with referents that on perceptual and operational grounds can be treated as singular — as in the appropriate pairing of "a lot" and "it" in the last sentence. With count-plurals the collective is figured more as an aggregate of discrete individuals:

- (30)There are coffees all over the far wall at the expresso bar.
- (31)She poured coffees from an ornate beaker.
- (32)There are a lot of coffees left on the table — shall I pour them out?

So mass versus count — the choice of which plural form to use — triggers an intrepretation shaping how the plurality is picturd and conceived (which is itself trigggered by the use of a plural to begin with). But if we restrict attention to just, say, count-plurals, there ar still different schemata for intending collections:

- (33)New Yorkers live in one of five boroughs.
- (34)New Yorkers reliably vote for Democratic presidential candidates.
- (35)New Yorkers constantly complain about how long it takes to commute to New York City.

The first sentence is consistnt with a reading applied to *all* New Yorkers — the five boroughs encompass th whol extent of New York City. The second sentence is only reasonable when applied only to the city's registered voters — not all residents — and moreover there is no implication that the claim applies to all such voters, only a something north of one-half. And the final sentence, while perfectly reasonable, uses "New Yorkers" to name a population completely distinct from the first sentence — only residents from the metro area, but not the city itself, commute *to* the city.

These examples demonstrate a point I made earlier, that mapping singular to plural is not a simple logical operation. We need to invoke narrative frames, interpretive scripts, and prelinguistic background knowledge to understand what *sort* of plurality the speaker intends. To be sure, the more subtle plurals can still be read in logical terms, and we can imagine sentences that hew more crisply to a logical articulation:

- (36)All New York City residents live in one of five boroughs.
- (37)The majority of New York voters support Democratic presidential candidates.
- (38)Many New York metro area residents complain about how long it takes to commute to New York City.

According to truth-theoretic semantics, sentences compl addressees to believe (or at least consider) logically structured propositions *by virtue of* linguistic shape replicating the architecture of the intended propositional complexes, as these would be represented in first-order logic. This view on linguistic meaning is consistent with the last three sentences, which are designed to map readily to logical notations (signaled by quasimathematical phrases like "the majority of"). But most sentencs do not betray their logical form so readily: these latter sentences actually sound less fluent, more artificial, than their prior equivalents.

It is also true that the more "logical" versions are mor, we might say, dialectically generalized because they do not assume the same level of background knowledge. Someone who knew little about New York geography could probably make sense of the latter sentences but might misinterpret, or at least have to consciously think over, the former ones. So we may grant that exceptionally logically-constructed sentences can be clearer for a broad audienc, less subject to potential confusion, and indeed such logically cautious language is a normal stylistic feature of technical, legal, and journalistic discourse. But for this reason such discours comes across as self-consciously removed from day-to-day languag. More important, the current examples show that if addressees *have* the requisite background knowledge, linguistic structure does not have to replicate logical structure very closely to be understood. The content which addressees undrstand may have a logical form, and language evokes this form — guides addressees toward considering specific propositional content — but this does not happen because linguistic structure in any precise way mimics, replicates, reconstructs, or is otherwise organized propositionally. Instead, the relation of language to predicate structures is evidently oblique and indirect: language triggers interpretive processes which guide us toward propositional content, but the structure of language is shaped around fine-tuning the activation of this background cognitive dynamics more than around any need to model predicate organization architecturally. In the case of plurals, the appearance of plural forms like "New Yorkers" or "coffees" compels us to find a reasonable cognitive model for the signified multitude, and this modl will hav a logical form — but the linguistic structures themselves do not in general model this form for us, except to the limited degree neeeded to activate prelinguistic interpertive thought-processes.

I make this point in terms of plural *forms*, and earlier made similar claims in terms of lexical details. A third group of triggers I outlined involved morphosyntactic *agreement*, which establishes inter-word connections which themselves trigger interpretive processing. Continuing the topic of plurals, how words agree with other words in singular or plural forms evokes schema which guide situational interpretations. So for instance:

- (39)My favorite band gave a free concert last night. They played some new songs.
- (40)There was some pizza earlier, but it's all gone.
- (41)There were some slices of pizza earlier, but it's all gone.
- (42)There were some slices of toast earlier, but there's none left.
- (43)There was some toast earlier, but they're all gone.
- (44)That franchise had a core of talented young players, but it got eroded by trades and free agency.
- (45)That franchise had a cohort of talented players, but they drifted away due to trades and free agency.
- (46)Many star players were drafted by that franchise, but it has not won a title in decades.
- (47)Many star players were drafted by that franchise, but they failed to surround them with enough depth.
- (48)Many star players were drafted by that franchise, but they were not surrounded with enough depth.
- (49)Many star players were drafted by that franchise, but they did not have enough depth.

Plurality here is introduced not only by isolated morphology (like "slices", "players", "songs"), but via agreements marked by word-forms in syntactically significant pairings: was/were, it/they, there is/there are. Framing all of these cases is how we can usually schematize collections both plurally and singularly: the same set can be cognized as a collection of discrete indic???uals one moment and as an intgral whole the next. This allows language some flexibility when designating plurals (as extensively analyzed by Ronald Langacker: see his discussion of examples like *Three times, students asked an interesting question*). A sentence discussing "slices of pizza" can schematically shift to treating the pizza as a mass aggregate in "it's all gone". Here the antecedent of "it" is *slices* (of pizza). In the opposite-drection, the mass-plural "toast" can be refigured as a st of individual pieces in "they're all gone". The single "band" becomes the group of musicians in the band. In short, how agreements are executed invites the addressee to reconstruct the speaker's concptualization of different referents discussed by a sentenc, at different parts of the sentence: linking *it* to *slices* or *cohort*, or *they* to *the band* or *the toast*, evokes a conceptual interpretation shaped in part by how morphosyntactic agreement overlaps with "semantic" agreement. Matching "they" to "the band" presents agreement in terms of how we conceive the aggregat (as a collection of musicians); using "it" would also present an agreement, but one schematizing other aspect of the band concpt.

In the last five above cases, "it" similarly binds (being singular) to "the franchise" seen as a single unit — here basic grammar and conceptual schema coincide — but "it" also binds to the "core of young players". The players on a team can be figured as a unit or a multiple. The franchise itself can be treated as a multiple (the various team executives and decision-makers), as in "they failed to surround the stars with enough depth". The last sentence is ambiguous between both readings: "they" could designate either the players or the franchise. Which reading we hear alters the sense of "have": asserting that the star *players* lack enough depth implies that they cannot execute plays during the game as effectively as with better supporting players; asserting that the *franchise* lacks depth maks the subtly different point that there is not enough talent over all.

The unifying thme across these cases is that when forming sentences we often have a choic of how we figure plurality, and moreover these choices can be expressed not only in individual word-forms but in patterns of agreement. Choosing to pronominalize "slices of pizza" or "cohort of players" as *it*, or alternatively *they*, draws attention to either the more singular or more multitudinal aspects of the aggrgat in question. But this effect is not localized to th individual it/they choice; it depends on tracing the pronoun to its antecdent and construing how the antecedent refrent has both individuating and multiplicative properties. Thus both individuation and plurality ar latent in phrases like *slices of* or "cohort of", and this singular/plural co-conceiving is antecedently figured by how subsequent morphosyntax agrees with the singular or, alternatively, the plural.

Moreover, these patterns of agreement invoke new layers of interpretation to identify the proper conceptual scope of plurals. In *The band planned a tour, where they debuted new songs* we hear the scope of "they" as narrower than its antecedent "the band", because only the band's *musicians* (not stage crew, managers, etc.) typically actually perform. Likewise in *The team flew to New York and they played the Yankees*, only the athletes are referenced via "they played" but presumably many other people (trainers, coaches, staff) are encompassd by "the team flew". And in *The city's largest theater company will perform* "*The Flies*" we do not imply that the Board of Directors will actually take the stage (the President as Zeus, say). Even in the course of one sentence, plurals are reinterpreted and redirected:

▾ (50)The city's largest theater company performed "The Flies" in French, but everyone's accent sounded Quebecois.

▾ (51)The city's largest theater company performed "The Flies"; then they invited a professor to discuss Sartre's philosophy when the play was over.

In the first sentence, the "space" built by the sentence is wider initially but narrows to encompass only the actual actors on stage. In the second, the "space" narrows in a different direction, since we hear a programming decision like pairing a performance with a lecture as made by a theater's board rather than its actors. I discussed similar modulation in conceptual schemas rlated to plurality and pluralization earlier; what is distinct in these last examples is how the interpretive processes for cognizing plurality are shaped by agreement-patterns (like *it* or *they* to a composit antecedent) as much as by lexical choice and morphology in isolation.

I have accordingly outlined a theory where lexical, morphological, and morphosyntactic layers all introduce "triggers" for cognitive processes, and it is these procsses which (via substantially prelinguistic perception and conceptualization) ultimately deliver linguistic meaning. What is *linguistic* about these phenomena is how specifically linguistic formations — word choice, word forms, inter-word agreements in form — trigger these (in no small measure pre- or extra-linguistic) interpretations. But as I suggested this account is only preliminary to analysis of how multiple interpretive processes are *integrated*. Linguistic *structure* contributes the arrangements through which the crossing and intersecting between interpretive "scripts" are orchestrated. Hence at the higher linguistic scales and levels of complexity, the substance of linguistic research, on this view, should gravitate toward structural intgration of intrpretive processes, even more than individual intrpretive triggers themslves.

This higher scale is my focus in the nxt section — seen from the perspective of formal and computational models as well as everyday language use.

Part III Procedures and Integration

So far I have criticized paradigms which try to account for linguistic meaning via concordance between linguistic and proppositional structur; the shape of predicate complexes. This critique has two dimensions: first, although a predicate structure, a predicative specificity, does indeed permeate states of affairs insofar as we engage thm rationally, such logical order is not modeled by language itself so much as by cognitive pictures we develop via interprtive processes *triggred* by language dtails but, I believe, to some not insubstantial degree pre- or extra-linguistic. Moreover, second, insofar as we *can* decelop formal models of language, these are not going to be models of prdicate structure in any conventional sense. Cognitive-interpretive processes may have formal structure — structure which may even show a lot of overlap with propositional forms — but these are not *linguistic* structures. Insofar as language triggers but does not constitute interpretive "scripts", the scripts themselves (i.e., conceptual prototypes and perceptual schema we keep intelectually at hand, to interpret and act constructively in all situations, linguistic and otherwise) are not linguiatic as such — and neither is any propositional order they may simulate. Language *does*, however, structure the *integration* of *multiple* interpretive scripts, so the structure of this integration *is* linguistic structure per se — and formally modeling such integration can be an interesting tactic for formally modeling linguistic phenomena. However, we should not assume that such a formal model will reemble or be reducible to formal logic in any useful way — formalization does not automatically entail some kind of de facto isomorphism to a system of logic (if not first-order then second-order, modal, etc.).

Instead, I want to focus in on branches of computer science and mathematics (such a process algebra, which I have already referenced) as part of our scientific background insofar as the *structural integration* of diverse "processes" (computational processes in a formal sense, but perhaps analogously cognitive processes in a linguistic sense) can be technically represented.

In "truth-theoretic" semantics, artifacts of language are intuitively pictured in terms of propositional structure. The guiding intuition compares word meanings to logical predicates; e.g. "John is married" is typically said when the speaker believes that, in fact, the predicate "married" applies to the speaker "John". Switching to a more "procedural" perspective involves intuiting word-maning more in terms of interpretive procedures than logical predicates. The change in perspective may not yield substantially different analyses in all cases, but I believe it does affect many cases.

Even a simple predicate like "married" reveals a spectrum of not-entirely-logical cases in ordinary language:

- (52)John is married to a woman from his home country, but he had to get the marriag legally recognized here.
- (53)John married his boyfriend in Canada, but they live in a state that does not recognize same-sex marriages.
- (54)John has been married for five years, but in many ways he's still a bachelor.
- (55)Every married man needs a bachelor pad somewhere, and wherever yours is, you need a mini-fridge.

We can make sense of these sentences because we do not conceptually define lexemes like "married" or "bachelor" via exhaustive logical rules, like "a bachelor is an unmarrid man". Instead we have some combination of prototypical images — familiar characteristics of bachelors, say, which even married men can instantiate — and a conceptual framework recognizing (and if needed distinguishing) the various legal, societal, and dispositional aspects of being married.

Intuitionwise, then, we should look beyond potential logical glosses on meanings — even when these are often accurate — and theorizee semantics as the mapping of lexical (as well as morphosyntactic phenomena) to interpretive scripts and conceptual or perceptual/enactive schema — which we can collectively designate, for sake of discussion, as "cognitive procedures".

The truth-theoretic mapping of words to predicates (and so phrasal and sentence units to propositional structures) provides an obvious way to formalize linguistic structur by borrowing the analogous structuration from predicate complexes. Substituting a procedural semantic model allows a comparable formalization of linguistic structure through theories exploring procedural integration, for instance the interactions between computational procedures. Analysis of *computational* produres can yield interesting idas for linguistic theories of *cognitive* procdures — without endorsing a reductive metaphysics of cognitive procedures as "nothing but" computational procedures implemented in some sort of mental software.

The notion of computational procedures is certainly not foreign to symbolic logic or to analyses oftn associated with logistical models (in linguistics and philosophy, say), like Typed Lambda Calculus. In those paradigms we certainly have, say, an idea of applying formulae to input values (where formulae can be seen as referring to or as compact notations for computational procedures). Comparing the results of two formulae yields predicates (e.g., equalities or greater/less-than relations of quantity) that undergirds predicate systems. Furthermore, combining formulae — plugging results of one formula into a free variable in another — yields new formulae that can in turn be reduced to different forms or to single values, either for all inputs or for particular inputs. In short, a large class of computational procedures can be modeled in predicate and equational or formula-reduction terms. I point this out because shifting to a "procedural" intuitive model does not *by itself* take us outside of logistic and truth-thoretic paradigms.

In computer science, theories of procedural networks begin to branch out from formal logic or lambda calculii when we move past the "mathematical" picture of procedures — as computations that accept a collection of inputs and return a collection of outputs — and theorize procedures mor as computational modules that input, output, and modify values from multiple sources, in different ways. In practical technology, this includes developments like Object-Oriented computer languages and programming techniques such as Exceptions and mutable references. Likewise, real-world cod can pull information from many sources — like databases, files, and networks (including the internet) — in addition to values input as parameters to a function. Conversely functions have many ways to modify values — many ways to manifest side-effects — beyond just returning values as outputs. Collectively these various channels of communication — wherein different procedures can exchange values in different ways — are often described as "inpure", by contrast to "pure" functions which use only a single input channel for (non-modifiable) input parameters and a single output channel.

The design and implementation of computer programming languages — sometimes called "Software Language Engineering" — is partly a theoretical discipline (because programming languages are based on formal systems that can be studied mathematically, like Lambda Calculus), but also very pragmatic. Computer languages are judged by practical considerations, like how efficiently they allow programmers to create quality software. The impetus for new programming techniques often comes from the practical side: mainstream languages began to incoporate features like Object-Orientation and Exceptions (which are a mechanism to interrupt normal program flow when coding assumptions are violated) because these features proved useful to programmers. At the same time, various "inpure" features can still be modeled at the theoretical level — there are extensions to Lambda Calculus with Exceptions and/or Objects, as well as "effect systems" that systematically model functional side-effects via Type Theory. In this case, though, the thoretical models are partly isolated or even lag behind the evolution of practical coding techniques.

It is also true that "inpure" code can always be refactored into a code base that uses only "pure" functional-programming techniques. That is, at least if we exclude the "cyberphysical" layer — the body of code that directly measures or alteres physical data, like writing text to a computer screen or reading the position of a mouse — any self-contained collection of "inpure" functions can always be re-implemented as a system of pure "mathematical" functions, using only one-dimensional, immutable input and output channels (this is not necessarily true of singl functions, but abstractly true of complete code *bodies* encompassing many functions, i.e., many implemented procedures). Analogously, any computing environment based on a type system that incorporates impure function-types can be recreated on top of a different type system that recognizes only pure functions. For sake of discussion, define a *procedure space* as a self-contained network of computational procedures, where each procedure has the ability to invoke other procedures in the network. We can then say that any procedure space developed with the benefit of "impure" coding styles (like Objects, Exceptions, and side-effects) is *computationally isomorphic* to a procdure space implemented exclusively in a pur-functional paradigm.

This principle might suggest that the pure/impure distinction is superficial, and so that any computing environment is actually manifesting a purely logical framework, because it is (in one sense) isomorphic to a structur which *can* be thoroughly modeled via formal logic. However, this reductionism actually reveals the limits of "computational" isomorphism in the abstract. Impure procedure-spaces are not *equivalent* to their pure isomorphs. The whole rationale for "impure" coding techniques — Object-Orintation, functions with side-effects, etc. — is that the resulting code better models empirical phenomena and/or how programmers themselvs reason about software design. Any notion of computational isomorphism which is *itself* prejudiced toward formal logic will fail to model how systems which are isomorphic *on these* (already logical) *terms* will nevertheless vary in how organically they model empirical and mental phenomena. If engineers discover that Object-Oriented paradigms produce more effective software for managing biomedical data — that is, biomedical concepts are usefully modeled in terms of semiautonomous computational "objects" whose properties and functionality are engineereed via many discrete, partially isolated software components — this suggests how Object-Oriented structures may be a more natural systematization of the underlying biomedical reality. Object-Oriented software can *in principle* be redesigned without Objects, but the resulting code base would thn be a weaker semantic "fit", a less faithful computational encoding, of the external information spaces that the software needs to modl and simulate.

I contend that this duality between Object-Oriented and pure-functional programming is analogous to the paradigmatic contrast between cognitiv and truth-theoretic semantics. On this analogy, pure-functional programming can be associated with truth-theoretic semantics in that both are founded on formal systems that more or less straightforwardly operationalize first-order logic — a logic with quantification and with sets or types modeling collections, but whose foundational units are boolean predicates and elementary formulae rather than proceures with multiple "channels of communication" and potentially many intermediate states. Reducing impure (e.g., Object-Oriented) code to pure-functional procedure spaces is analogous to providing truth-theoretic reconstructions of linguistic meanings, as in (revisiting examples from last section):

▾ (56)People had fed the rescued dogs (i.e.: before they were rescued).

▾ (57)New Yorkers vote democratic (i.e.: the majority of registered voters in New York do so).

Propositional reconstructions of sentences lik thse can capture a logical substrate that contributes to their meaning, but completely *reducing* their semantics to such logical substrata jettisons, I contend, interpretive and contextual schemata which ar equally essential to their meaning. Analogously — considered in relation to data that it must curate and model (so that pople may manage and aggregate data; for instance, collect biomedical data to ensure proper diagnoses and implement tratments) — software does not only represent the logical substrata undrlying information systems. It needs to capture conceptual and operational/logistical phenomena as well.

Along these lines, note that Object-Oriented techniques originally emerged from scientific-computing environments, to model complex systems, with multiple levels of organization, mrgent structure, and so forth (in the 1990s these techniques were then applied to more enterprise-style applications, for User Interface design, business operations workflows, etc.). The effectiveness of Object-Oriented models suggests that as a form of computing environment, and a framework for coding procedure-spaces, Object-Orientation captures structural properties of complex systems more richly than paradigms that reduce procedural implementations to a purely logical substratum. This observation can then generalize to the whole range of modern-day programming techniques and all the various domains where software is used to monitor, simulate, and manage information about natural and human phenomena.

Technically, the multi-dimensional structure of computational procedures implmented with modern coding techniqus — multi-dimensional in the sense of multiple forms of channels of communication and functional side-effects — demands new concptualizations of the formal architecture of computing nvironments as such. Rather than seeing computer software, for example, as a complex system built architecturally from predicate logic, we need to undrstand software as a procedure-space or procedural network receptivvto external information. On this account, a software application internally possesses a set of capabilities, and achieves this functionality by invoking different procedures which are implemented in the software; the application as a whole is the aggregate of its procedural building-blocks. Which procedures get invoked depends on user actions — we use a mouse, keyboard, and other input devices to trigger responses from the software. Applications are therefore built around an "event loop": the software can enter a passive state, performing no operations apart from monitoring physical devices to detect user actions. Onc an action does occur — we press a letter on a keyboard, say, or press a mous button — the application responds by invoking a specific procedure, which in turn may trigger a complex cascade of other procedures, to complete whatever operation we requested via our action (saving a file, selecting one record to visualize from a database, etc.). Via such networks of procedures and functionality, software models and simulates empirical phenomea — at least enough so that we can maintain databases, remotely operat machines, and in other ways track and manipulate physical states.

Consequently, computer models of empirical phenomena can take the form of *procedural networks*, and we can designate the design and theory of such models as *procedural modeling*. I have argued that procedural networks are not reducible to logical predicate systems even though procedure-spaces in general are (in a certain sense) isomorphic to procedure-spaces which are more purely logical, or employ pure-functional type systems. The "isomorphism" relevant here, I am arguing, eliminates semantic and simulative structures that make procedural models effective constructions conveying the organization and behavior patterns of complx, real-world phenomena. On this theory, procedural modeling is a more effective tool for representing real-world systems than are computing environments built more mchanistically from predicate logic.

Interestingly, however, the predominant contemporary paradigms for modeling real-world information systems — particularly what has come to be called the "Semantic Web" — is built on a framework of description logic and "knowledge graphs" rather than (at least except very indirectly) anything that could be called "procedural data modeling". There is, of course, a robust ecosystem of network-based code-sharing that provides a viable infrastructure for a more procedural data-sharing paradigm. To illustrate the contrast, consider the problem of biomedical data integration: of merging the information known to two different mdical entrprises, as when a patient moves between two different hospital systems. We can assume that each hospital uses a different set of software tools to store patient records and manage data from their various secondary or constituent sources (diagnostic labs, specialized departments, outside clinics, etc.). Such multi-faceted data then needs to be translated from formats native to the prior hospital's software to formats needed by the new hospital.

Such data-intgration problems tend to be concptualized in one of two ways. One approach is to seek a common reprsentation, or an overarching unified model, which can reconcile structural differences between the two systems. If both hospitals use the same biomedical "Ontologies", for example, then those Ontologies serve as a logical paradigm through which the distinct systems can be bridged. In effect, structural differences between the systems are treated as superficial variations on a common or unified logical "deep structure" — analogous to translating between natural languages by mapping sentences to a propositional core. Indeed, the field of "Ontology Engineering" can be seen as a way of marshaling abstract logic into a form that is practically useful for information storage and extraction, in open-ended environments where data may be aggregated from many heterogeneous sources. The term "Ontology" in this context is not wholly unrelated to its philosophical meaning, bcause "knowledge engineers" assuming that the primordial structuring paradigms of such "universal" logical systems are relations and distinctions investigatd by the philosophical ontological tradition — substance and attribute, part and whole, time-point vs. time-span, spatial region vs. spatial boundary, subtype and supertype, and so forth. Abstart Ontological systems are then formalized into logical rules that provide an axiomatic core — sometims called "Upper" Ontologies — which are then extended via empirical models or "domain-specific" rules to create Domain-Specific Ontologies used to integrate data in concrete enterpris/scientific fields (medical information, government records, and so forth).

The overarchong paradigm of such Ontology-based integration is the idea of a logical model that recognizes superficial differencs between incompatible (or at least not fully compatible) ata sources. An alternative approach to data integration problems is to treat these as issues of procedural capability. In order to import data from one hospital into a second hospital system, for example, assuming their respective software and databases are at least somewhat incompatible, you need to perform certain procedures that translate data from the first format to the second. Insofar as this is possible, the two software systems gave a certain procedural synergy: the capabilities of the first system include exporting data in a format the second can use, and the capabilities of the second system include integrating data presented in formats that the first can export. This synergy does not need to be theorized in terms of an overarching logical ur-space which both systems indirectly manifest; instead, it reflcts how some subset of the overall procedural network germane to each respctive software system includes — either by deliberate cooperative enginering or because the two systems are guided by similar standards and technical orientations — procedures on the two ends that can be aligned in terms of the kind of data and data-formats they produce and/or consum. In short, targeted (and potentially cooperatively-engineered) procedural alignments — rather than overarching logical commonalities — form the bedrock of data integration.

These contrasting paradigms are not only theoretical. Computer scientists hafve actually tried to promote data sharing and data integration to improve governance, health care, urban infrastructure (via "smart cities"), sustainable development, and so on. Insofar as we see data integration in terms of Ontology Engineering, these practical goals can be met with the curation and publication of formal Ontologies that can guide software and database development. Data sharing is then driven by Ontological alignment — the Semantic Web, for example, designed as an open repository of raw data annotatd and structured in accord with publishd Ontologies, data which can bereusedand integrated into different projects more readily because it adopts these published models. On the other hand, insofar as we see data sharing in terms of *procedural alignment*, we prioritize th curation and publication of procedure implementations, in the form of software libraries, open-source code repositories, and other building-blocks of modular software design. Consider again the case of data integration between two hospitals: to integrate data between the two, programmers may need to implement special "bridge functions" that reconcile their rspective formats. This implementation is simplifid insofar as the respective software systems are well-organized and transparent — so that programmers can examine inport and export functions in the two code bases and write new procedures, extending thir respective capabilities, so that import functionality on one end can be aligned to export functionality on the other. These new procedures can then be shared so that similar data integration problems — for instance the first hospital sharing data with a third — can be solved more readily, reusing some of the code thereby developed. This approach to data integration emphasizes transparency, modularity, and code-sharing.

Rather than seeing the Semantic Web as a network of raw data conforming to published Ontologies, the more procedural perspective would see endeavors like a Semantic Web as driven by code sharing: open repsoitories of procdural implementations that can be used to reconcile data incompatibilities, pull data from different sources, document data structure and representation formats, etc. Decentralized networks like the Semantic Web would then be characterized by the free exchange of procedural implementations, so that enginers can pull procdures providing different capabilities togther, to assemble fully-featured software platforms. Code libraries would play a homogenizing role in this paradigm analogous to Ontologies in the Semantic Web. And, indeed, there exists a mature and sophisticated technical infrastructure for publishing software componenrs and maintaining code repositories, forming the productive underbelly of the Open-Source ecosystem ("git" version control, the GNU Compiler Collection, Linux distributions, etc.). However, the Semantic Web and Open Source development communities are largely separate, apart from the practical given that many Semantic Web technologies are distributed as Open-Source software. Despite the *practical* adoption of Open-Source norms, the Semantic Web community has arguably not engaged the Open-Source community at a deeper theoretical level, in terms of how th curation of public, collaborative code libraries can promote data integration analogous in effect to (but arguably semantically more accurate than) Semantic Web Ontologies — in light of critiques that the logical intuitions behind the Semantic Web create a distorted and oversimplified theory of what semantics is all about (in the words of Peter Gardenfors, "The Semantic Web is not very Semantic").

These questions bear directly on Phenomnology, because philosophers in the phenomenological tradition have directly influenced the evolution of the Semantic Web — notably Barry Smith, who has both published sophisticated theoretical work on Ontologies in the Semantic Web sense and also spearheaded practical initiatives like the OBO (Open Biological and Biomedical Ontology) Foundry. The OBO Foundry emerged in the mid-2000s, on the heels of a renewed interest in Phenomnology as a philosophical basis for Cognitive Scienc and other practical/technical disciplines, like knowledge engineering. It is not hard to see practical artifacts like the OBO system as concrete realizations of theoretical goals articulated in volumes such as 1999's *Naturalizing Phenomnology*. This association is not only at the level of scholarship — the academic papers describing OBO, for example – but also the design and organization of Semantic Web tools like the OBO Foundry, as technologies and platforms.

Seen in those terms — and if we restrict attention to work done by scientists like Barry Smith and his colleagues who are actively enagaged in both the phenomenological and computer-science communities — Semantic Web technology suggests a paradigm wherein "Naturalizing" Phenomenology involves isolating logical structures which are at once essential to modeling empirical data and also emerge organically from phenomenological accounts of perception and cognition — that is, capturing the logical order of our experiencing the world as well as of the facts experienced. A case in point is formalizing systems of "mereotopology" — combining the mereological account of part vs. whole with "topological" models of spatial continuity, locality, and bounday — which reflect both perceptual schema and information gestalts. So one the one hand the fusion of mereological and topological relations gives us a vocabulary for describing how we experientially apprehend spatial forms — the continuity, regionality, intersections, and disjunctions between visual (and sometimes tactile) elements that gives logical articulation to visual/tactil sense data (never experienced as "raw" sensation sinc we fundamentally experience space and visual continuity in these structured forms). Meanwhile, on the other hand, Mereotopology provides a semantic matrix for representing facts and relations in biological, geographical, and other scientific data, so it serves a practical information-management role. In short, isolating logical gestalts and then codifyung them in practically useful forms serves to "Naturalize" Phenomenology by anchoring phenomenological reflection in applied science.

The general implication of this "Naturalizing" strategy is that Phenomenology becomes naturalized, or reconcild with the physical sciences, insofar as structures of consciousness can be aligned with logical systms. This strategy seems to extend beyond just the Semantic Web projects I have highlighted; it is likewise evident for example in Kit Fine's logical mereology in Barry Smith and David Woodruff Smith *Cambirdge Companion to Husserl* ("Part-Whole", pp. 463). Indeed Husserl himself invites us to consider the logical formalization of phenomenological systems in works like the *Formal and Transcendental Logic*, which appears to suggest that human conceptual systems are an even more refined manifestation of logical systems than are "formal" logics which get entangled in model-thoretic problems like the Lowenheim-Skolem thorem. In other words, systems of formal logic are codifications of a mental order and therefore natural cadidates for a technical representation of the mental realm in its structure and specificity.

However, we can also observe that Husserl was writing at a time when abstract logic was still the preeminent phenotype for systematic exposition of formal structures in general — this was before computer programming and even before mathematical developments like Category Theory. In the first half of the last century, someone hoping to create a formal and systematic representation of cognitive processes would gravitate toward symbolic logic simply because mathematicians and philosophers at the time followed the general intuition that *any* formal structure was essentially characterized by its logical/axiomatic foundations. A century later, formal logic has been displaced from the germinal origins it was once assigned. For mathematicians, logic itself is revealed to be a kind of emergent system that depends on Categorial definitions like limits and colimits, and can vary across Categories — there are different logics for different kinds of Categories. As such it is not logic itself but the properties and contrasts between Categories which is the truly primordial foundation of logico-mathematical thought. A contemporary logico-mathematical formulation of Phenomenology may therefore try to establish a Category-Theoretic grounding rather than a logical one as ordinarily understood. A case in point would be how Jean Petitot situates phenomnological analysis in certain soecific genres of Categories, like sheaves and presheaves, in his "Morphological Eidetics" chapter in *Naturalizing Phenomenology* and elsewhere.

Petitot's and Barry Smith's formalizing projects were parallel and collaborative to some extent. Maxwell James Ramstead in a 2015 master's thesis reviews the history elegantly: "Now, the "science of salience" proposed by Petitot and Smith (1997) illustrates the kind of formalized analysis made possible through the direct mathematization of phenomenological descriptions. Its aim is to account for the invariant descriptive structures of lived experience (what Husserl called "essences") through formalization, providing a descriptive geometry of macroscopic phenomena, a "morphological eidetics" of the disclosure of objects in conscious experience (in Husserl's words, the "constitution" of objects). Petitot employs differentiai geometry and morphodynamics to model phenomenal expenence, and Smith uses formai structures from mereotopology (the theory of parts, wholes, and their boundaries) to a similar effect." Except, there are interesting contrasts between the Cognitive-Phenomenological adoption of mereotopology (by Barry Smith and also Roberto Casati, Achille C. Varzi, Kit Fine, Thomas Bittner, etc.) — which stays within a more classical logical paradigm — and Petitot's Morphological Eidtics, which is more Category-Theoretic.

Meanwhile, contemporary formalizations of phenomenological analyses can also gravitate toward a more concrete and computational framework — simulating cognitive processes via software or comparing artificial constructions of perceptual objects (via Virtual Reality, 3D graphics, 3D printing, robotics, etc.) to lived exprience. In particular, graphics engineers have a rich theory about how to create realistic (albeit not truly life-like) 3D models and scenes. The mathematical and computational elements in this theory — triangular, quadrilateral, and polygonal meshes; shader algorithms; "NURBS" (Non-Uniform Rational Basis Spline) surfaces; textures and *uv*-mapping; camera matrices; diffusion and stochastic processes — create a formal model of perceptual phenomena insofar as these can simulated *to a close approcimation*: visual phnomena artificially built with these techniques can be *almost* realistic. The gaps between such "virtal" scenes and real life may suggest that there are additional facets suffusing "ral life" perception, or even that despite their realism the mathematical building-blocks of CGI-like scenes are fundamentally different from the formal structures governing nurophenomenological perception. But in any case the almost life-lik realism that *can* be achieved via Computer Graphics is a data-point that Phenomenology should acknowledge: that a perceptual world built out of certain rigorously mathematical constitunts can feel almost lifelike when apprehended as if it were a real visual-perceptual surrounding. In particular, Computer-Generated Imagery can evoke the same embodied engagement and intentional patterns as non-artificial, ambient perception so long as we accept a certain "suspension of disbelief", or mentally adjust to the phenomenological limitations of seeing visual tableau on a two-dimensional screen — analogous to watching movies or television. Despite the phenomenological chiasma of directing visual attention to a 2D screen — it feels "not quite right" — we can still become engaged and largely immersed in the visual scenes before us; which means that full phenomenological realism is not prerequisite for our intentional comportment toward visual (and auditory) phenomena. We can then observe that constructed scenes built entirely from mathematical carry comparable potential for intentional engagement. Moreover, Panoramic Photography and Immersive Visual Reality present another genre of phenomenological immersion that transcends the limitations of the 2D screen — though still without full realism, because however lifelike the visual content we may perceive with, say, 3D goggles, we still are not engaging tactilely and kinaesthetically with the world in the usual ways.

In short, one route toward a formal framework for elucidating perceptual-phenomenological content is to examine realistic simulations of visual contents as computational artifacts — not in terms of abstract formulations (logical or otherwise) but in terms of concrete computer code and software. With reference to Merotopology, for example, we can contrast the logical groundwork set out by Barry Smith (for example) with the differential-geometric and category-theoretic landscape considered by Jean Petitot *and also* a more "experimental", software-driven intuition associated with, for example, Virtual Reality research. Looking at mereotopology in particular, this more computational approach can examine how part-whole relations are created within CGI and Computer Aided Design by mesh alignment or texture mapping, or how texture and diffusion algorthms create effects of material continuity and locality. In this case mereotopological notions are not embedded in logical systems — or even, from a computational perspective, in formal Ontologies — but rather latent in graphics code.

For scholars pursuing a "Naturalized" Phenomenology, then, we (in the 2010s and 2020s) have several avenues to choose from, including logical formalism (including as practically leveraged in the Semantic Web) but also mathematical formalization (as with Category Theory and Differential Geomtry) and, also, computationally. Computer Aided Design and Computer Generatd Design point to a theory of formal structures producing life-like (if not perfectly realistic) perceptual content. Analogously, computational models of linguistic structure can help represent the organizing principles of our reasoning toward language understanding — even if these formal models are not proposed as direct simulations of natural linguistic cognition.

Simultaneously, real-world applied projects — like the Semantic Web, Virtual Reality, or CGI/CAD technology — can be seen as practical test-beds for the realism and analytic potential of formal-phenomenological frameworks. In some cases, like the OBO Foundary, the link between applied technology and Phenomnology is explicit; in others, as with VR and CGI, this link is more implicit and thematic (but still addressed by interdisciplinary research grounded in Phenomenology and Husserl scholarship). But in any case technological experience retroactively seems to shape the direction of phenomenological research, while at the same time Phenomenology has, for some researchers, provided a metatheortic and metaphysical guideline.

Here the Semantic Web presents an interesting case-study, because the manifestation of phenomenological themes (e.g., Mereotopology) in a practically useful resource like Biomedical Ontologies suggests an *ex-post-facto* vindication of *logical* formalization as a "Naturalizing Phenomenology" project. On the other hand, critiques of the Semantic Web — which have emerged, among elsewhere, from Cognitive Linguistics — can accordingly be studied as potntial indications of how classical log formalism is limited in the phenomenological context. Peter Gardenfors, for example, has critiqued the Semantic Web on thoretical grounds while also developing a model of Natural Language semantics (via a thory of "Conceptual Spaces") that we may find more phenomenologically realistic than the Semantic Web's (as Gardenfors puts it) "syllogistic" paradigms. In this paper, I propose a critiquefrom a more "procedural" angle. From my perspective, the foundational characteristic of "information systems" is the existence of *procedures* (say, cognitive and/or computational procedures) which "act on" (aggregate, interpret, reshape) the data at hand. I believe it is a fair critique to say that Semantic Web technology has unduly discounted the procedural dimension of information management — not only in a thoretical sense, but also quite practically. For example, the OBO Foundary does not include a mechanism for code sharing with the goal of curating software libraries that implement atatypes conformant to the various published Ontologies. In the Semantic Web paradigm, defining logical formalizations of standardized concept-systems is considered orthogonal to implementing software components where these formal criteria ar eralized in practic. In short, *logical specification* is trated as distinct from *implementation*. This is not a universal approach: many trvhnological standards are published at least in part via "r"eference implementations which demonstrate standardized concepts and guide other implementations, helping to enforce compatibiloty btween different software components. Moreover, code sharing and code reuse can promote interoperability no less effectively than alignment relative to logically defined standards. So there *are* technological trends that emphasize "procedural alignment" and code-sharing as important contributors to data integration. However, these branches of technology do not appear to have exerted a strong intuitive influence on the Semantic Web.

I argued above that technology has a retroactive influence on Phenomenology, or at least on the threads of research that follow the "Naturalizing" project and the reconciliation of Phenomenology with Analytic Philosophy. Even if this effect is rather modest, it still bars on the topic that is my primary emphasis here, namely the integration of Phenomenology with Cognitive Grammar. One of the most prominent practical domains that has influenced Phenomenology has been the Semantic Web, insofar as Semantic Web technology (via Ontologies) show some evidence that formal models influencd by Phenomenology can be practically useful, which is one criteria to suggest that the models have philosophical or epistemological merit. On the other hand, Cognitive Linguists have tended (if anything) to be critical of the Semantic Web (in contrast to other linguistics branches, which are generally sympathetic to Semantic Web paradigms and incorporate Ontologies into Computational Linguistics software). So a potentially fertile ground for collaboration between Phenomenology and Cognitive Linguists has arguably been overlooked insofar as the respective communities have taken competing lessons from th succsses and limitations of the Semantic Web, particularly how the Semantic Web leverages classical logical formalisms (particularly Description Logics) rather than "procedural" and/or Category-Theortic foundations.

This problem is not intrinsic to the Semantic Web as a data-sharing platform, however, only to the paradigms through which the current Semantic Web has been conceptualized and implmntd. There are competing intellectual frameworks that embrace parallel goals but present alternative technical foundations, and Phenomenology would benefit from thematizing these frameworks in a role analogous to the Semantic Web, both a practical application and a retroactive intuition-guide. A case in point would be the OpenCog project, that presents a Hypergraph-based modeling paradigm related but technically distinct from Semantic Web labeled graphs (one which is also consistent with procedural-network models) and which also embraces a specific linguistic model (based on Link Grammar). Philosophically, OpenCog appears to celebrate a vision of "Artificial General Intelligence" which I (and I suspect most phenomenologists would) find problemmatic; underestimating the context-sensitivity and empathic intersubjectivity intrinsic to human cognition and intelligence (and difficult to simulate with machins). Nevertheless, formal models designed to *replicate* intelligent behavior can still be useful as structural *models* of cognitive phenomena, even if we believe in a metaphysical gap between the model and the reality. Implementing software on the basis of explanatorarily useful cognitive models does not guarantee that the software will realistically approximate human cognition, but articulating and fine-tuning the models themselves can notwithstanding be a valuable exercise.

In the case of OpenCog, the impetus toward "Artificial General Intelligence" has motivated that project to explore cognitive models that coalesced around several key structures, including Directed Hypergraphs, Link Grammar, and (in my terms) Procedural Networks (as unerlying mols for Information Spaces). This aggregate of theories can be juxtaposed to Description Logic, Directed Labeled Graphs, and formal Ontologies as the groundwork for the Semantic Web. I'd also argue that the OpenCog model can potentially be a technological improvement over the existing Semantic Web, in the sense that semantic networks built around OpenCog-like structures can be more effective vis-à-vis several important practical concerns, like application design, data integration, and Human-Computer Interaction. These comments do not necesarily apply to the actual OpenCog software — the major OpenCog component, "AtomSpace", is in a practical sense harder to compile and use than most Semantic Web components — but instead to a potential standardization of the core OpenCog data structures as modeling paradigms that can be adopted by heterogeneous information sources and data-sharing initiatives. In this eventuality, the OpenCog model can provide a test-bed for applied Phenomenology that stands on a different formal foundation — one less inured to symbolic logic.

Along these lines, then, I contend that a circle of data models analogous to the OpenCog architcture can provide a formal structuration for phenomenological accounts of linguistic processing and information spaces comparable in analytic roles to 3D modeling primitives in a Phenomenology of Perception. On one side, mathematical elements like mesh geomtry, NURBS surfaces, and texture mapping/generation point toward formal theories of perceptual *cognition* by allowing for the construction of perceptually realistic *scenes*. These mathematical elements are building-blocks for an (artificial) perceptual *content* rather than (as far as we know) actual neurocognitive subvenants of perceptual *experience*, but their formal specificity still gives us material to work with when trying to consolidate a "scientific" phenomenological research programme. Analogously, I suggest that OpenCog-like structures such as Directed Hypergraphs, Procedural Network models, and Link Grammar are potential building-blocks for formal models of Cognitive Linguistics and "information management” — for our procssing of both linguistic content and the contxts and situations wherein language artifacts are grounded. In other words, these procedural/hypergraph/link-grammar structures can model linguistic “deep structure” and linguistic environments by analogy to how mesh geometry, texture mapping, and so forth model 3D spatial primitives and visual-perceptual environments.

This idea is still somewhat hypothetical because the “procedural/hypergraph/link-grammar” nexus has not been consolidated into a general to the same degree as a common vocabulary of 3D modeling primitives has been incoporated into disparate software and research projects. Having said that, Link Grammar itself does have standing as a distinct and institutionally circumscribed body of research, so it is a reasonable starting point for integration with phenomenological and cognitive-linguistic approaches, an integration which can then be extended to related structures like Procedural Networks and Directed Hypergraphs.

For such reasons, I will conclude this paper with a focus on Link Grammar and on how the Link Grammar perspective relates to the phenomenological and cognitive-grammar positions and case studies I outlined in earlier studies. Further extensions to Directed Hypergraphs and the technical foundations of multidimensional “procedural” type theories defined on Directed Hypergraphs — which thematically complement Link Grammar to the degree that Link Grammar and Directed Hypergraphs are naturally paired up as in OpenCog — are beyond the scope of this paper (though I have presented some form of this anlysis in other publications).

Part IV Procedural Networks and Link Grammar

My goal in this section is to incorporate Link Grammar into a phenomenological and Cognitive Grammar perspective, more than to offer a neutral exposition of Link Grammar theory. Therefore I will accept some terminology and exposition not necessarily originating from the initial Link Grammar projcts (though influenced by subsequnt research, e.g. [expectation]). I also want to wed Link Grammar to my own semantic intuitions, set forth earlier, that word-meanings and morphosyntactic interpretations should be grounded on pre- or para-linguistic cognitive “scripts” that are activated (but not structurally replicated, the way that according to truth-thoretic semantics linguistic form evokes-by-simulating propositional form) by linguistic phenomena.

Link Grammar is, depending on one’s perspective, either related to or a variant of Dependency Grammar, which in turn is contrasted with “phrase structure” grammars (linguists tnd to designate competing schools with acronyms, lik “DG” for Dependency Grammar and “HPSG” for Head-Driven Phrase Structure Grammar). Link and Dependency Grammars define syntactic structures in terms of word-pairs; phrase structure may be implicit to inter-word relations but is not explicitly modeled by DG formalisms — there is typically no representation of “noun phrass” or “verb phrases”, for example. Phrase structure is instead connoted via how relations fit together — in “rescued dogs were fed”, for instance, the adjectival “rescued”-“dogs” relation interacts with the “dogs”-“fed” (or “dogs”-“were” plus “were”-“fed”) predication, an interaction that in a phrase-structure paradigm is analyed as the noun-phrase “rescued dogs” subsuming the noun “dogs”. Dependency analyses often seem more faithful to real-world semantics because, in practic, phrases do not *ntirely* subsume their constitunt parts. Linguistic structure is actually multi-layered, where semantic and morphosyntactic connections resonate between units within phrases separate and apart from how linguistic structure is organized into phrasal units themselves.

Except for phrases that coalesce into pseudo-lexemes or proper names (like “United Nations” or “Member pf Parliament”), or indeed become shortened to single words (like “waterfall”), we perceive phrases both as signifying units and as aggregate structures whose detailed combinative rationale needs contectualization and interpretation. In short, phrases are not “canned” semantic units but instead are context-sensitive performances that requir interpretive undrstanding. This interpretive dimension is arguably better conveyed by DG-style models whose consituent units are word-relations, as opposed to phrase-structure grammars which (even if only by notational practice) give the impression that phrases conform to predetermined, conventionalized gestalts.

While Link and Dependency Grammars are both contrastd with phrase-structure grammars, Link Grammar is also distinguished than mainstream DG in terms of how inter-word relations are conceived. Standard DG recognizes an assymetry between the elements in word-relations — one element (typically but not exclusively a word) is treated as "dependent" on another. The most common case is where one word carries greater information than a second, which in turn adds nuance or detail — say, in "rescued dogs" the second word is more essential to the sentence's meaning. This potentially raises questions about how we can actually quantify the centrality of one word or another — in many cases, for instance, the conceptual significance of an adjctive is just as trenchant as the noun which it modifies. In practice, however, the salient aspect of "head" vs "dependent" assymetry is that any inter-word pair is "directed", and one part of the relation defined as dependent on another, however this dependency is understood in a given case.

By contrast, Link Grammar dos not identify a head-dependent assymetry within inter-word relations. Instead, words (along with other lexically signifant units, like certain morphemes, or punctuation/prosodic units) are seen as forming pairs based on a kind of mutual incompleteness — each word supplying some structural or signifying aspect that the other lacks. Words, then, carry with them different varieties of "incompleteness" which primes them to link up with other modls. Semantic and grammatical models then revolve around tracing the *gaps* in information content, or syntactic acceptability, which "pull" words into relation with other words. This approach also integrates semantic and syntactic details — unlike frameworks such as Combinatory Categorical Grammar, which also treats certain words as "incomplete" but identifies word connctions only on surface-level grammatical terms — Link Grammar invites us to explore how semantic and syntactic "completion" intersects and overlaps.

Words can be incomplete for different reasons and in different ways. Both verbs and adjectives generally need to pair up with nouns to form a complete idea. On the other hand, nouns may be incomplete as lexical spotlights on the extra-linguistic situation: the important point is not that people feed dogs in general, but that *the rescued* dogs were fed prior to their rescue. So "dogs" "needs" *rescue* for conceptual specificity as much as "rescue" needs "dogs" for anchoring — while also "dogs" needs *the* for *cognitive* specificity, because the discourse is about some prarticular dogs (presumed known to the addressee), signald by the definitive article. In other cases, incompleteness is measured in terms of syntactic propriety, as in:

▾ (58)We learned that people fed the rescued dogs.

▾ (59)No-one seriously entertained the belief that he would govern in a bipartisan manner.

In both cases the word "that" is needed because a verb, insofar as it forms a complete predicate with the proper subject and objects, cannot always be inserted into an enclosing sentence. "People fed the rescued dogs" is complete as a sentence unto itself, but is not complete as a grammatical unit when the speaker wishes to reference the signified predicate as an epistemic object, something believed, told, disputed, etc. A connector like "that" transforms the word or words it is paired with syntactically, converting across part-of-speech boundaries — e.g. converting a proposition to a noun — so that the associated words can be included into a larger aggregate.

The interesting resonance between Link Grammar and Cognitiv Grammar is that this perspective allows us to analyze how syntactic incompleteness mirrors semantic incompletenss, and vice-versa. "Incompleteness" can also often be characterized as *expectation*: an adjective "expects" a noun to produce a more tailored and situationally refined noun (or nominal "idea"); a verb expects a noun, to form a proposition. Analogously, when we have in discourse an adjctive or a verb we expectv a corresponding noun — so via syntactic norms language creates certian expectations in us and then communicates larger ideas by how these expectations are met. Is a noun-expectation fulfilled by a single noun or a complex phrase? The notion of smantic and syntactic expectations also coordinates nicely with type-theoretic semantics; for example, the verb "believe" pairs with a semantic unit that can be interpreted in epistemic terms — not any noun but a noun of a kind that can be the subject of propositional attitudes (beliefs, opinions, assertions, arguments, etc.).

The syntactic incompleteness of propositional phrases modified by "that" can therefore be traced to the semantic expectations raised by "believe", and analogous verbs (opine, argue, claim, testify). Th object of *testify*, say, is a statement of potential fact which we know not to take as necessarily true or honestly made (part of the nature of testijony is that it may be deliberately or accidentally fallacious). But to properly pair with "testify", then, phrases must be semantically reinterpreted as nominalizations of propositions, rather than as mere linguistic exprssion of propositional content via complete sentences. The "epistemic" context transforms sentential contnt into nominal contnt available for further refinement:

▾ (60)The Trump campaign colluded with Russia.

- (61)Several witnesses testified that the Trump campaign colluded with Russia.
- (62)Reputable newspapers have reported that the Trump campaign colluded with Russia.
- (63)Most Democrats believe that the Trump campaign colluded with Russia.

The grammatical stipulation that a modifier like "that" is often necessary in such formulations correlates with the semantic detail that the "claimed", "testified", or "believed" content is not being directly asserted by the spaker as if in a unadorned declarative expression, as in the first sentence.

Morphosyntactic transformation similarly modls th corrlation btween semantic and syntactic expectation — as can be demonstrated by a variant of the "believe" forms, via the phrase "believe in":

- (64)I believed in Father Christmas.
- (65)I believed in Peace on Earth.
- (66)I believed in Obama.
- (67)I believed in lies.

Whereas "that" (after "believe") "nominalizes" propositions, "in" reconceives (type-theoretically we would say "coerces") ordinary nouns into epistemic nouns (compatible with propositional attitudes). Obama is not an *idea*, but the connector *in* triggers an interprtation where we have to read "Obama" as something believed — effectively a type-theoretic tension resolved by understanding "Obama" in this context to designate either his platform or his ability to implement it. Intrpretive *tension* is a natural correlary to a mismatch in xpectations: "believe" expects something epistemic, but the discourse gives us a proper name. Analogously "budge" expects a brute physical ntity in its simplest meaning, but in "Obama wouldn't budge on reproductive rights" we get a "sentient" noun, and have to read "budge" metaphorically. In short, *expectation*, *interpretive tension*, and *incompleteness* are interlocking facets of semiotic primitives that gestate into discursive maneuvers via which ideas are communicated economically and context-sensitively.

Link Grammar, proper, represents only the most immediate (mostly grammatical) facet of word links. For sake of discussion, I will discuss links in general as markers of *mutual dependency* between words, so a "link grammar" is essentially a "bi-directional" Dependency Grammar. Mutual dependencies manifest syntactic norms and contextual details that make individual words inadequate signifying vehicles for a particular communicating content. This overall principle becomes concrete in one form via grammatical relations, which is the layer modeled by Link Grammar proper. I have mentioned several ready examples — the syntactic dependency of verbs and adjectives on nouns for them to enter correctly into discourse (correlate with a semantic dependency in the other dirction, to shape noun-ideas to the proper context and signifying intent); also part-of-speech or "subtyping" dependencies reflecting mandates that (in my examples) propositional phrases are coerced to nouns or nouns coerced to "epistemic" nouns. Technical Link Grammars recognize a broad spectrum of "link relationship" — between 50 and 100 for different languages. Parsing a sentence is then a matter of identifying all of the mutual dependencies — at least those evident on a syntactic level — that appear as inter-word links in the sentence. Phrase structure may be implicit in some links in combination — for example verb-subject plus verb-object links generate propositional phrases — but the technical parse is a "graph" of inter-word links rather than a "tree" of phrases ordered heirarchically. The parse-graph itself is only a provisional rading of the sentence, and linguistic understanding exists only insofar as its skeletal outline is filled out with semantic and situational details. But the graph layer articulated by a Link Grammar still provids a useful inermediat representation, showing mutual dependencies in their syntactic manifestations that then point toward thir deeper semantic and situational origins.

For each *syntactic* bi-dependnecy, on this theory, there is a concordant semantic and signifying bi-dependency, partly conventionalized as a feature of the language and partly hewn to the current discourse context. To leverage Link Grammar in an overall Cognitive Linguistic environment, then, we need to examine the semantic relations that drive syntactic bi-dependency: how the grammatical structure of one word completing another is a codification of of *semantic* mutual dependency. The *semantic* bi-dependencies operate on both abstract and concrete levels. Abstractly, it is obvious that an adjective or a verb dependents on a linked noun to complete an idea. This abstract prototype of bi-dependency then takes concrete forms in each specific discourse, acquiring detail and specificity from situational contexts.

The crucial dimension in this theory is neither abstract nor concrete bi-dependency but the intersection of the two. The conventionalized lexical, syntactic, and morphosyntactic norms of a language present abstract prototypes of mutual word dependencies. The concrete instantiation of these forms — via word-pairs whose surface presentation indicates the presence of specific link relations (often with the aid of morphology, agreement, spoken inflection, and other morphosyntactic cues) — invites us to consider how an abstract bi-dependency becomes situationally concretized in the present, momentary context. In essence, abstract mutual dependencies are the raw materials from which situational appraisals are creates. A pairing like "rescued dogs" uses a certain abstract-bidependent prototype — here the double-refinement of a noun and adjective grounding each other — to trigger the listener's awareness that the spakr's discourse is centered on one specific aspect of the dogs (that they were at some point rescued) with its concptual corrolaries and unstated assumptions (that, being in need of rescue, they were abandoned, in danger, and so on). Similarly the further link to the definit article — "*the* dogs" — evokes the prototype of a definite article grounding a noun, which in turn communicates the speaker's beliefs about the current state of the discourse.

This last "bi-dependency" deserves further comments, because nouns more often than not reveal some dependency on an article: "some dog(s)", "the dog(s)", "a dog", "many dogs". These pairings paint a picture both through the choice of article and the presence or absence of a plural. This picture is partly situational — obviously whether the speaker is talking about one or multiple dogs is situationally important — but it is also meta-discursive: selecting the definite article indicates the speaker's belief that the listeners know which dogs are on topic. The lexeme "the" thereby signifies a meta-discursive stance as well as a cognitive framing — both that the collection has enough coherence to function as a conceptual unit in context, as *the* dogs (and not something less specific like "*some* dogs") *and also* that the speaker and listeners share compatible cognitive pictures as a result of prior course of the discourse. This also introduces several avenues for future discursive evolution — the listeners can respond to the speaker on both cognitive and meta-discursive levels. A direct question like *which dogs?* signifies that the first speaker's meta-discursive were flawed — the referent of "the dogs" has not been properly settled by the discourse to that point. Or questions for clarification — like *how many dogs are there?* — indicate the listeners' sense that all parties' respectiv construal of the situation needs to be more neatly aligned for the discourse to proceed optimally.

The point I particularly want to emphasize here, though, is that these discursive/cognitive effects inhere not only in the word "the" but in its pairing with other words, like *the dogs*. We tend to see the lexical substratum of a language as the ground level of its signifying potential, but we should perhaps recognize bidependent prototypes as equally originating. Th communicative content of "the dogs" is ultimate;y traced not only to the lexical potentialities of "the" and "dogs" as word-senses, but to the abstract prototype of the definite-article bidependence, which becomes concretized in the "the dogs" pairing at the same time as the individual words do.

In order to bring this account full-circle, I would then add that lexical units mutually completed by an instantiated bidependence can also be seen as a tie-in between interpretive procedures. Lexical interpretive scripts — the cognitive processing solicited by "the" and "dogs" in isolation — are themselves open-ended and un- or incompletely grounded. We can speak metaphorically of "words" being incomplte, or carrying expectations, but it is really the cognitive scripts associated with words that are lacking in detail. The resolution of a merely schematic cognitive outline to a reasonably complete situational construal can be likned to a rough sketch filled out in color — but we have to imagine that a sketch can be completed by pairing it with a second sketch, and the content in each one, crossing over, allows a completed picture to coalesce. "The" in itself evokes an interpretive process that in itslf cannot be completed, and likewise "dogs"; but each script supplies the content missing from the other. In this sense the bidependent form concretized by the pair is actualy evoking an interpretive phenomenon of mutual completion — the language structure here is guiding us toward an intrpretive interpenetration where the two scripts tie each other's open ends. Whereas lexical items can be associated with single "scripts", prototypes of mutual dependency model patterns in how script-pairs can become mutually complete. But unlike lexemes, which are notated directly by language, the instantiation of bidependnet script-pairs occurs indirectly. Some paired-up words are of course adjacent, but adjacency between words does not have the same binding determinacy as sequencing among morphemes in *one* word. Instead, word adjaceny is only one signal among many others suggesting that some prototypical inter-word relation applies between two words (which might be widely separated in a sentence). Morphology and syntax also point towards the pairings operative in a sentence — they are to bidependency prototypes what word-choice is to the lexicon.

Thus far I have made an admittedly *philosophical* and speculative case for "interpretive mutual dependence" as a constituent building block of linguistic understanding. This theory will remain troublingly incomplete if the more philosophical presentation cannot be wed to a more rigorous formal methodology. True, an essential core of this theory is that interpretive "scripts" are largely prelinguistic and so not covered by linguistic formalisms in themselves. However, I have also argued that formal linguistic structures *do* govern how we identify which links apply to which word-pairs and the gneral outlines of how word-pairing coordinats cognitiv processes associated with single words — the fully contextualized synthesis of lexically triggered cognitive procedures may involve extra-linguistic grounding, but abstract prototyps of bidependnet relations are also prototypes of a synthesis between cognitive/interpretive functions. It would accordingly be reassuring if notions like "bidependency" and "mutual completion" could be employed as foundations for a formal theory of grammar and/or semantics with a degree of rigor comparable to, say, Link Grammar in its computational form, or type-Thoretic Semantics in the sense of Zhaohui Luo, James Pustejovsky, etc... Such a theory — and potentially concrete technologies associated with it — would also then have a reasonable ground of comparison to the Semantic Web and, in the contxt of Phenomnology, to the formalizing influenc which Semantic Web paradigms have exerted on projects to unite Phenomenology with science and with Analytic Philosophy.

Given these considerations, I propose that formal grammars with the same underlying structure as Natural-Language Link Grammars can indeed be used as a foundation for type-theortic and programming-language-design methodology. The key step here is to generalize Link Grammar's notion of a "connector" — the aspect of a word or lexeme that allows (or requires) compltion via another word — to a gneric data structure where connections can be made between different parts of a system on the basis of double potentials that must be in some sense "compatible" for the connection to be valid. One way to visualize such a systm is via graph thory: imagine a form of graps where nodes are annotated with "potentials" or "half-edges"; a complete edge is then a union of two half-edges. Half edges are also classified into different families, and there are rules governing when a hald-edge of one family may link with a half-edge of another. In the case of Link Grammar, these classifications are based on surface language structure — head/dependent and left-to-right relations — from which a suite of links and connection rules are define (for instance abstractly a head/right word must link to a dependent/left word, a rule that then becomes manifest in specific syntactic rules, like how a verb links to its subject). For a more generic model, however, we can stipulate only that there is *some* classification of connectors governs by *some* linkage rules, to be specified in different details for different modeling domains.

Such a graph model expands upon the notion of *labeled* graphs, where edges are annotated with labels that characterize the kind of relation modeled via the edge itself. A canonical example is Semantic Web graphs: the edges in any Semantic Web structure are labeled with "predicates", defined in different Ontologies, specifying what sort of relation exists between its adjacent nodes. That is, in the Semantic Web, nodes are not "abstractly" linked but rather exhibit concrete relations: a person is a citizen of a country, two persons are married, and so forth. These structurs ar ethen concete instancs of Labeled Graps as abstract mathematical structures. Based on Link Grammar, we can then refine this modle be splitting labels into two parts, and allowing edges to be incomplete: a fully formed edge is possible when the label-parts on one side are compatible with the label-parts on another. One valid class of graph transforms is then a mapping where a graph is altered by unifying two half-edges into a complete edge, subject to the relevant linkage rules.

Another way of modeling this kind of structure is via edge-annotations and a rule for unifying two edges into an edge-annotation pairing. For sake of discussion, I will express this in terms of Directed Hypergraphs: assume that edges are "hyperedges", connecting *sets* of nodes. In Hypergraph theory, the nodes incident to a hyperedge are divided into a "head set" and "tail set"; these sts can then aggregate as "hypernodes". We can then define a kind uf unification where the "tail hypernode" of one hyperedges joins with the "head hypernode" of another, producing a new hypredge whose head comes from the first former hyperdge and whose tail comes from the secon. The merged hypernodes, in turn, form a new hypernode which "annotates" the new hyperedge (this new hypernode is not connected to the graph via other nodes, but is indirectly "part" of the graph through the hyperedge it annotates). *Annotated* hyperedges therefore differ from "non-annotated" hyperdges in that the former are the result of a merger between two of the latter. The rules governing when such merger is possible — and how to map a pair of hypernodes into a single "annotative" hypernode (which belongs to the graph through the aegis of its annotated hyperdge) are not internal to the graph theory, but presumed to be specified by the modeling environment where implemementations of such graphs are technologically applied. Annotated Directed Hypergraphs are then "complete" in a sense if every "un-annotated" hyperedge has been subsumed into an annotated hyperedge, via a fusional process we can call a "annotative-fusional transform".

Extending this model further, we can say that the tail of an *un-annotated* hyperedge is a "tail pre-annotation", since it is poised to be merged into an annotation. Analogously, the head of an un-annotated hyperedge is a "head pre-annotation", and "annotative fusion" is the synthesis of a head and a tail pre-annotation (triggering a synthesis of their incident hyperedges). Correlate to annotative *fusion* we can define a notion of annotative *partiality*, referring to the "incompleteness" of pre-annotations which leaves room for their fusion.

It turns out that annotative fusion and partiality in this sense is a non-trivial model for computation in general, and can be extended to a form of type theory and process calculus. The idea is that computational procedures can be modeled as hypergaphs (computer source code can certainly be modeled as hypergraphs which are productions of a certain class of parsing engines). Each "value" affected by a procedure — or more technically the source code symbols and memory addresses that "carry" a value — is then modeled as a hypernode that can link with other hypernodes in the scenario where one procedure calls a different one. Annotative fusion is then a phenomenon of values being transferred from one excution environment (associated with the caller procedure) to a second one (associated with the callee). The "annotations" themselves are then in this context the full set of type coercions, type checks, synchronization (e.g. resource locks or thread blocks depending on whether or not the caller waits for the callee to finish), and any other validations to ensure that the procedure call is appropriate. Annotative fusion also provides a formal basis for developing the intuition that "procedural networks" are rigorous representations of information spaces — annotative fusions capture the precise details of procedures linking (via caller/callee relations) with other procedures.

The constituent units of procedural networks are inter-procedure calls — but procedural networks also reveal dimensions of connectivity and and clustering characteristic of large, complex networks (and the graphs that represent them) in general. What appears as one function-call in source code can actually represent many different inter-procedure connections, a phenomenon reflecting "overloading" and "genericity" in programming language theory. Functions are generic in the sense that any one of their arguments can take multiple types — either because the function is explicitly declared to tak a "typeclass" or a single type for that argument, or because an instance of a given type may actually be at runtime an instance of some subtype. The engine which actually implements inter-procedure calls — i.e., the programming-language implementation — needs to factor this genericity into runtime decisions, so a single expression in one function body can branch to many different called procedures. This is the essential core of the "semantics" of programming languages: data structures manipulated by computer code do not *intrinsically* represent real-world, non-digital phenomena, though they ar enginered to model external data when used properly. However a code base does *internally* posess a space of implementd functions, and a symbol at one place in source can match to some set of other functions so as to effect a procedure call. This "matching", and the rules governing how "overload resolution" occurs — "overload" meaning that a given notated procedure call can actually branch to multiple functions, so runtime information is needed to select the right one — are the essential formal principles governing the semantics of computer code.

From this basis, all the same, computer code can model a wide range of empirical phenomena. Generic code represents generic patterns of functional organization, allowing models to be built from varying layers of abstraction. From this perspective, to describe an empirical system it is neessary to identify important behaviors and functional patterns via which the system's observed behavior can be notated and/or simulated. To the degree that systems take on functional organizations that can be abstractly described, similar to the functional dynamics of other systems, their behavior can be represented and/or simulated via gneric cod. To the degree that thre are particular details of a system's behavior that are more idiosyncratic to that system, and need to be modeled precisely, procedures can be crafted specifically for obsevring and emulating that exact behavior. More generic and more exact procedural implementations can coexist in a single code base, with generic functions calling granular functions narroed to precise types, and vice-versa. The coexistence of generalization and specificty is an essential fatures of code bases and, by extension, of procedural networks, ensuring the flexibility of these ntworks as tools to model information spaces.

Unfortunately, this kind of "procdural" modeling is hard to intgrate with the more static techniques reprsentd by the Semantic Web and the current "Big Data" fad. The latter paradigms tend to treat data as a static repository to be mined for patterns and insights, rather than a digital simulation or encoding of dynamic real-world systems. The Semantic Web, as a large, collaborative modeling project, evolved largely apart from the technological community concerned with computer simulations and the programming techniques which emerged from there, like Object-Orientation. This divergnce is relevant for linguistics and cognitive science, because I would argue that the more "dynamic" paradigm is actually more "Semantic" in a Natural Language sense. In other words, our cognitive dispositions when interpreting empirical phenomena — and matching these interpretations to linguistic cues — are more like procedural networks capturing functional patterns and layers of genericity in observed phenomena, rather than an accretion of static data. The techniques of procedural data modeling may therefore be relevant for Cognitive Linguistics and Cognitive Phenomenology because they aspire to something which, arguably, the mind does instinctively: build cognitive or computational models of dynamic, functionally organized phenomena.

As a corrolary, the theoretical building-blocks of Procedural Data Modeling — how it leverages type theory, programming language semantics, and so forth — can provide at least analogs or case-studies for corresponding cognitive phenomena. Here I would argue that generalizing Link Grammar from Natural Language to formal languages, type systems, and lambda calculii yields added structures to type thory that are useful toward a more rigorous "theory" of Procedural Data Modeling — a thory of natural linguistics generalized to a theory of gneral data reprsentation which, in turn, may offer insights onto the cognitive dynamics underlying (prelinguistic) situational/perceptual comportments and interpretation.

Type-theoretically, annotative partiality — which recall is my terminology for the abstract generalization of the mutual "incompleteness" in Link Grammar connctors, driving their link-fusion — extends conventional applied type theory (as in the Typed Lambda Calculus) in parallel to partial-labels extending labeled graph theory. It is paradigmatic in the theory of typed procedures and of "effect systems" that the type of a procedure is determined (up to certain equivalences that may discard overly granular type distinctions) by the types of all values affected by the procedure (including but not necessarily limited to the types of input and output parameters). We can then superimpose on this model an account of annotative partiality. Specifically, on the paradigm that procedure-calls are structurally represented as annotative fusions over Directed Hypergraphs, the values manipulated by a procedure are pre-annotations: they are not (in the *implementation* of a procedure, as a formal object) single values but rather typed spaces that can take on a spectrum of possible values depending on the inhabitants of their types. When a procedure is *called* these values become concretized, but as a formal system procedural networks model software in terms of its capabilities and expected behavior, rather than the state at any moment when the software is actually running. Partiality therefore models how precedures (as formal objects) deal with potentialities — we do not know what values will *actually* be present at runtime (e.g. what specific values passed to a procedure as arguments), so procedural analysis is essentially characterized by a partiality of information. When one procedure calls another, the caller must build an *expression* — a gathering of values that provide all the information the callee requires — thereby creating a case of mutual-completion: the caller has values but not an algorithm to operate on them; the callee has an algorithm (that's what it implements) but needs concrete values so to produce concrete values. This dual partiality allows the caller to call the callee, via an *expression* which is part of the callee's implementation (represented as a hypergraph) which must in turn match the callee's signature — epigrammatically, we can say that "expressions are annotative-fusional duals of signatures". The point here is that whereas signatures are conventionally understood to be type-declarations assigning types to procedures, with annotative partially we can more precisely recognize signatures as stipulating *pre-annotative* types. The values carried within expressions also have pre-annotative types, but there is a distinction between types in the context of expressions and types in the context of signatures — and moreover this distinction is precisely the manifestation of the abstract head-pre-annotation and tail-pre-annotation contrast in the specific context of procedural networks. Just as signatures unify multiple types into one profile, we can analogously define "expression types" as the aggregate of all types from values affecting the expression — note that this is different from the type of the exprssion's calculated *result*, just as the type of a function is different from the type of its return value. Expression-types and signature-types are almost exact duals (the complication being default values for optional parameters, which are not directly represented in exprssions — obviously, since then they would not be missing). The "duality" involved here derives from partitioning a type system into "expression annotative partials" and "signature annotative partials", a projection of head/tail duality in an abstract theory of Annotated Directed Hypergraphs (and analogous to head/dependent and left/right partiality in Link Grammar).

This notion of expression/signature duality is not just speculative; I believe it has rich potential for new techniques in data modeling, program analysis, and code verification, and is a core dimension of software distributed as part of a research-sharing platform, currently used in connection with several publications spanning multiple publishers and academic disciplines.[1] The path I have sketched here from Link Grammar to a theory of Procedural Networks is similar, if not technically equivalent to, analogous analyses associated with OpenCog, for example (particularly in Ben Goertzel's two-volume *Engineering General Intellignce*).

The software I just dscrribed in rlation to certain published data sets certainly is narrower and more nascent than OpenCog components like AtomSpace. At the same time, it has been more directly formulated in relation to academic and publishing workflows and is more precisely sited in academic and research-sharing domains (and moreover remains agnostic vis-à-vis the more extravagant "Artifical General Intelligence" claims). It is also almost entirely self-contained, with few external software dependencies, which arguably makes this software a more accessible test-bed for refining and

[1] See Rodger2016, Prague2015, BHP2018, Neustein2019, and the "ScignScape" repository on github: most of these are examples where authors or publishers supplemented a prior book or article with a curated data set, or "Research Object", where the data set is downloaded along with code from which customized software can be compiled specific to the data set and text being shared, software I devloped in part as a prototype or experimental sandbox for computational models mentioned here (in the case of the 2019 volume, part of Springer's "Advances of Ubiquitous Sensing in Healthcare" series, the data sets and publication are developed in consort).

analyzing formal properties of structures I have outlined here — in particular a generalization of Link Grammar to a working model of computation, perhaps via what I have dubbed "annotative fusion". In any case, both OpenCog and, I would like to believe, this published software (designed to emulate some of OpenCog's architectural features but on a smaller-scale and mor self-contained foundation) are concrete technologies that take Link Gramar and Procedural Networks (in one sense or another) as modeling and linguistic paradigms. In this sense they may be contrasted with Semantic Web technologies, which are based on labeled directed graphs and Description Logics. Phenomenologically, we can then give formal analogs to back up the comparisons I intimated earlier: between the Semantic Web as a formal case-study and retroactive intution for Phenomenology, but one paradigmatically grounded in classical logic and truth-theortic semantics, and other paradigms — here developd via Annotated Hypergraphs, annotative fusion, and annotative partiality add a new structural layer to lambda-calculus-style type thory — informed via linguistic (rather than logical) methods like Link Grammar and arguably more compatible with Cognitive Grammar (and Cognitive Phenomenology). Here I make the implicit assumption that such an "annotative partiality" and hypergaph-based type theory prsents a formal foundation fundamentally different, in some philosophically interesting sense, from classical formal logic. Fully defending this claim is outsude the scope of this paper, but I will offer a few comments on this topic in conclusion.

Part V Conclusion

In the last section, I outlined a theory of *annotative fusion* which generalizes (and migrates from natural to formal languages) Link Grammar to fields like type theory, programming language design, and (suitably extended) lambda calculii. The crux of this extension is taking a formulation of hypergraph transforms — where hypergraph edge-pairs map to annotatd hypredges — and then reading "annotations" as general computational structures, where an "annotative fusion" represents the executive machinery requisite to implement (sequential or concurrent) calls between procedures. As with link grammar, which recognizes many link types (or in these terms many premises for tail-to-head pre-annotation fusion), this rough picture allows many different kinds of inter-procedure calls, coding paradigms, and functional side-effects to be represented. For example, Object-Oriented code exhibits a particular kind of inter-procedure connection, subject to special semantic and syntactic conventions (notably the object value in an expression is bound to a "this" or "self" symbol in the callee code). In languages like C++ and Java, Object-Oriented expressions are notated by placing the object symbol (or nested expression) before a "dot" token and then a function name. The "dot" then signals the presence of two connectors, one to its left that marks a symbol or expression as an object (and thus connected to a "this" symbol in callee code) and one to its right that marks a symbol as a function name (and so bound to whichever procedure uniquely resolves the procdure-call at runtime given the runtime type of the arguments).

As this example shows, one could write a formal grammar for Object Oriented languages — e.g. for a valid expression using an object and a valid signature of a function that takes an object as its "this" argument — with syntax-rule representations essentially identical to Link Grammars for Natural Language. In place of link-types like verb-subject, verb-object, antecedent-pronoun, etc., this *programming* language grammar would have connections like object-expressions and object-symbols to "this" tokens, thrown exception-exprssions to symbols declared in "catch blocks", and so on. Expanding this idea, we can say that modern programming languages have multiple form of inter-procedure exchange of values, and that for each such alternative there is a distinct link-type and a distinct pair of "connectors" (in established Link Grammar terminology) or "pre-annotations" modeling how values get passed between procdures insofar as that communicative protocol is in effect. Object-Oriented code uses one protocol for objects, a different protocol (essentially the classical Lambda Calculus) for other functional arguments, and a third protocol for throwing and catching exceptions. In the context of this paper, these protocols correspond with specific compatibilities between pre-annotations and specific regimes for annotative fusion. Within each experession/signature match, there are in general multiple "annotative fusion regimes" which must all be properly provisioned by the desired procdure-call for the fusion to be acceptable. The values subject to any particular regime I then collectively call a *channel*, and the rules for validating fusions I refer to as a "Channel Algebra". Type declarations on procedures are then enriched with the added structure of multiple "annotative fusion regimes" (which can also be called "channel prototypes", or "proto-channels") along with the distinction between expression-pre-annotation and signature-pre-annotation types I mentioned earlier. These added structures add levels of detail to the basic notion of "functional types" in the Typed Lambda Calculus, and we can define type systems that are closed over simpler functional type systems with respect to annotative-fusion, annotative-partiality, and proto-channel operators. In Neustein19 I describe these enriched theoris as "Channelized Type Systems" and give a more expansive outline and explanation of their formal properties. For the moment, I will just claim for sake of discussion that "Channelized" type theory according to these criteria is a reasonable extension to clasical type theory and that the published software represents a concrete implementation of a prototype formal language whose type system is Channelized in this sense.

Here I openly acknowledge that I am not providing a rigorous mathematical exposition of these Channelized Type Systems, analogous to how other extensions would be presented in the type-theory community proper. This is partly be personal choice: I am both more interested in and more capable of examining forms of type theory via their concretization in implemented programming languages, rather than their mathematical exposition in "abstract" programming languages. It is certainly possible to formulate formal languages as mathematical constructs, using theorems as de facto guarantees of their behavior. Such hypothetical languages are still not actually implemented, however. Provable claims about programming languages are importamt, but also valuable are specific implementations of languages — i.e., software that reads source-code files conformant to the language grammar and executes operations described by that code — partly because the code realizing the language itself then becomes a concrete artifact which can be studied and analyzed in turn.

Concrete language implementations are therefore an alternative framework for developing type theory, separate from purely mathematical type theories and from the kinds of type systems central to mathematical foundations research (e.g., Homotopy Type Thory). I'll refer to type theory in the former sense as *experimental*, in comparison to "logico-mathematical" type theories, on the basis that claims about the type systems of concrete programming languages are often defended by appeal to empirical observations about program-behavior rather than formal proofs.

I have concluded this paper with a brief reference to a "Channelized Type Theory", which I turned I arived at in several phases, from Phenomenology to Cognitive Grammar to Link Grammar to Annotated Directed Hypergraphs. As with any juxtaposition of philosophical/speculative thories and formal/technical ones, claims of inter-disciplinary unification (or even overlap) need to be cautious and provisional. It certainly is not self-vident that Channelized Type Theory (or Link Grammar and Dircted Hypergraphs, or Semantic Web Ontologies and Dircted Labeled Graphs) represents an adequate or relevant formalization of phenomenoligical structures. To the degree that Channelized Type Theory (by analogy to Semantic Web paradigms) can be retroactivly interesting to Phenomenology at a systematizing or intuition-honing level, the resonances between the two need to be dilated from well-established theortical overlaps (like Phenomenology to Cognitive Grammar), tracing a pivot to increasingly formal and/or technological structures (as with formalizing Cognitive Grammar via Link Grammar and then generalizing Link Grammar to formal languages and type theories). This paper has been able to present only a few examples tieing the phenomenologuical side of this progression to the technological side at each step along the way.

But without presuming that the mutual relevance between Channelized Type Theory and Phenomenology has been demonstrated here in anything but skecthy detail, we can still consider the meta-philosophical situation if such a connction can be made even provisionally. As I have proposed, Channelized Type Theory (at least the way I have developed it) is "experimental" rather than rigorously logico-mathematical; as a result, a potential formaliztion for some phenomenological structures is being presented as a technological and esxperimental or empirical artifact rather than an abstract mathematical system. My interest in formal models for Phenomenology is inspired by the "Naturaliing Phenomenology" project and particularly by Jean Petitot's formalizing studies (alongside David Woodruff Smith's metaphysical analyses which, I think, provide a persuasive philosophical grounding for Petitot's "eidetic" scientization). However, Petitot's interdisciplinary orientation is centered on identifying mathematical structures (like sheaf mereology) which seem to replicate some of the cognitive patterning of perceptual and situational experience; in short it is formalization via mathematical spaces. As I intimated when contrasting Petitot with Barry Smith, the former's strategy is still a departure from "logical" formalizations — sheaves are Categorical phenomena rather than set-thoretic ones, so sheaf theories are not reductively conflated to logical systems (though like any Category sheaves "have" a logic in some sense). Ptitot's analyses, in short, are formalizations of certain phenemenological gestalts via mathematical categories rather than via the formulation of logical systems that would purport to simulate phenemenological structures via axioms or syllogisms (again as a counter-example I would cite Kit Fine's mereology based on Husserl's *Logical Investigations*).

But relative to Petitot's mathematical treatments, I would argue that "experimental" formalisms are even more significant departures from a philosophical paradigm which essentially reduces formalization in general to a presentation of logical-axiomatic reconstructions of empirical (emergent, higher-scale) phnomena. The rationale behind exercising a new type theory via concrete software exemplifications rather than via a theorem-driven mathematical treatment is that essential structural details may be evident in th concrete implementation which get lost in a mathematical reduction. After all, the principle of "Turing equivalnce" sugests that at some mathematical level all programming languages are behaviorally indistinguishable. Insofar as grammatical or pragmatic differences between programming languages are nonetheless interesting as case-studies for semiotic or philosophical themes — insofar as programming languages, while "constructed" and constrained by the need to manipulate computer operations as well as express coding intent for the benefit of other programmers, are still "languages" and can suggest insights for human languages as well — the fact that higher-level programming structures get reduced away under mathematical treatments of programming languages shows that treating these languages as just mathematical objects is not a useful theoretical maneuver in this contexts. After all, several high-level features of some programming languages — including Object-Orientation, monads, functional side-effects, continuations, type coercion, and lambda abstraction — have been studied as potential formal models of Natural Language features as well. These analyses clearly require and presuppose a theory of programming languages where Objects, monads, effect-systems, and the like are first-class posits rather than epiphenomena that get analytically erased under mathematical reduction.

But such a non-reductive theory of programming languages arguably needs to be "experimental", centered on these languages as software artifacts, whose properties and behavior are tested empirically. If progreamming languages are also case-studies for Phenomenology, their status as formalizations is likewise experimental — Phenomenology grounding itself in a science that is observational rather than logico-mathematical; that progresses by practical tinkering rather than by asserting axioms and proving theorems.

This experimental modus operandi does not only affect how research proceeds; it also alters how research intersects with the txtual and publishing ecosystm. If computer code is an essential part of an autgor's argumntation, then the author is responsible for curating a code base as well as perfecting a written exposition. The publication itself does not only exist as a book or article but extends also to an open-source repository which needs tobe used and maintained. The author herself presents as competent in the technological and computational skills associated with creating and maintaining an open-source repository as well as in the subject-matter covered in academic text. Disciplinary expertise is complemented with literacy in practical tools needed to work with repositories, like GitHub and the GNU Compiler Collection.

There is also a more subtle sense that curating a complex code base demands — or brings forth — an experimentalist attitude that stands apart from the conventional style of philosophical thought. Software development is unapologetically trial-and-error, ad-hoc, intuitive, relativly informal — the programmer is a hacker as much as a scientist. In the landscape of 20th-century philosophy, software development is metaphorically post-modern, perhaps Deleuzian. It reaches asymptotically toward formal precision — this is a precondition of any commitment to software quality — but tries to achieve formal guarantees through a crucible of trial-and-error and test-and-failure. The programmer who writes software as part of a research project is suspended between two different regimes of scientific rigor, or communicative rationality — the software itself is (albit in digital incarnation) a *thing*, which hoperfully does what it is supposed to, a mechanical ventriloquist endorsing th author's claims. What software offers the research ecosysem is a raw performativity that transcends disciplinary boundaries. We can dispute the author's claims and how much leeway we grant the author in defending these claims, but not the behavior of the software itself. The author's "hacker" persona lifts her outside of any institutional or disciplinary enclosure other than the push to get *this* code to work, this one further piece to the puzzle. But any claim that software — apart from its practical uses — by its very implementation serves as a kind of warrant for scientific or philosophical claims brings us back to a communicative and disciplinary rationality — the onus is on the author to justify that her software's behavior demonstates some principle of acadmic interest, something outsid itself. In my case the relevant claim would be, for example, that the code paired with this paper demonstrates how Channelized Type Theory or "annotative fusion" is a credible and practically useful model of computation.

But at least when evaluating these claims we're not just talking about philosophical arguments and the intellectual ecosystms where philosophical discussions happen — the journals, classrooms, and conferences. We're also talking about software and technology — something that can be downloaded, experimented with, improved, fully transparent. The ad-hoc rationality of this transparency is somewhat different than the partial obscurity of peer review, closed-access journals, monetized conferences, or disciplinary exclusivity. Software is not a degree. This transparent rationality is perhaps a natural complement to Phenomenology's "to the things themselvs". Software is, par excellence, expertise in the thing itself. Software is intellectual cryptocurrency; the intelligence of the digital challenging the rigor and scientificity of the brick-and-mortar and the institutional. As the mortar unifying a research community software is syndicalistic rather than hierarchical. But in this sense collaborative software development — as well as the communal evaluation of published software — is an intriguing instance of Habermassian discourse communities. Software can help lead philosophy back to its Habermassian ethical wellspring.

Perhaps ironically, the thread developing Phenomenology in its most “Naturalizing” and scientific aspirations is also a gateway to an experimental, even “post-modern” and even political and politicized temperament. The Analytic and the existential-political trajectories of Phenomenology are not so far apart, after all.