Vajra Logic and Mathematical Meta-models for Meta-systems Engineering

ON THE FOUNDATIONS OF HOLONOMIC METASYSTEMS THEORY AND ENGINEERING

Kent D. Palmer, Ph.D.

P.O. Box 1632
Orange CA 92856 USA
714-633-9508
~~palmer@exo.com~~, kdp@exo.com

Version 0.01; 11/20/01; vl01a01.doc

Keywords: Meta-systems Engineering, Systems Engineering, Mathematical Models, Meta-models, Schemas, Diamond Logic, Vajra Logic

Introduction

A 'system' is a particular kind of schema that we project on things. We need to recognize that there are other kinds of schemas such as 'meta-system' or 'domain' or 'world' that are also projected on things. The schemas form an ontological emergent hierarchy[1] that is opposite the ontic emergent hierarchy[2] discovered in things. This difference is celebrated as the dualistic distinction between logos and physus within our Western worldview. This leads us to understanding that what is needed is a General Schemas Theory which explains the nature of the emergent hierarchy of schemas that we project on things and its difference from the organization of the things themselves at the various levels of emergence. This need is particularly poignant in the case of Systems Engineering Design in which the ontological emergent schemas are used as internal archetypal blueprints that are a basis for producing products that change our world; that is products that have emergent properties. The question that arises is how do we ground this production that we already engage in but do not understand completely ourselves. How do we produce systems that have emergent properties? How do these systems fit in to the context and content of the other schemas within the hierarchy of emergent schemas? This becomes particularly important when we realize that we have been ignoring to our own peril the other broader schemas like meta-system, domain and world. The terrorist incident of September 11th 2001 shows that others are able to intervene within our technological infrastructure at the level of these broader schemas to do us harm. Suddenly it becomes important to begin to design the higher level schemas themselves, not just designing the systems and ignoring

---

[1] The ontological emergent hierarchy might be:
pluriverse
kosmos
world
domain
meta-system
system
form
pattern
monad
facet

[2] The ontic emergent hierarchy might be:
social group
animal
organ
multi-cell organism
cell
macro-molecule
molecule
atom
particle
quark
string

their interactions and side-effects, designing the meta-systemic environments, designing the domains, designing the worlds we inhabit as well. In this way new disciplines called meta-system engineering, domain engineering and world engineering come to the fore as needing to be articulated. We have been implicitly engaged in these broader levels of design for a long time. Now we need to include them explicitly in our compass of what constitutes "systems engineering", now thought of more properly as Schema Engineering.

This paper deals with the grounds of this new discipline in which we consider a different hierarchy that takes from our design theory up to the paradigm, and then the episteme to the level of ontos. Any particular design theory uses schemas in order to give it internal coherence. Normally we think of it as patterned data contents encapsulated by objects (forms plus behavior) that makes up a system. We design the system to produce some emergent qualities that would be useful to support our intentions within our world. What we normally fail to do is to take into account the side-effects of these designed systems within the world and extend our design vision beyond the system into the meta-systemic environment, to the domain and world levels of organization. Recently we have begun to speak of systems of systems in order to indicate a broader perspective. But this term merely reiterates the schema of the system at a higher level of abstraction rather than recognizing the fundamental difference between the system and the meta-system.

When we begin to think about the differences between the system and its complementary inverse dual, the meta-system, we are suddenly in a kind of foreign territory. We tend to want to ground our systems thinking in mathematics and construct formalisms which explain the nature of the system in terms of parts and relations between these parts. This does not explain the wholeness of the system that Rescher points out in his work Cognitive Systemization that is based implicit models of the organism. If we look to systems theory we notice that analytic definitions of the system schema prevail, such as that of Klir in his key work Architecture of Systems Problem Solving. Klir defines for us a discipline independent model of the formal structural system, i.e. a unified approach to things that combines the schemas of pattern in terms of structure, form and system. What is needed is a similar combination of the schemas meta-system, domain and world which would give us an articulation of the context within which formal structural systems arise and interact. Here however we will concentrate on the grounding of the meta-system because it is the next step in the broadening of our conception of the task of systems engineering.

In order to ground the meta-system we need to understand the way in which theories depend on paradigms which in turn depend on epitomes and finally depend on ontologies. Our systems designs are theories that we test by bringing the systems they blueprint into existence and placing them into our world to see how they operate within that world. These design theories are based on schematic paradigms which give them internal coherence. We talk about paradigm shifts when our assumptions behind our theories change. But what is not normally mentioned is the fact that these assumptions that produce the paradigm our theories are based on have to do with our schemas, i.e. the inner coherence of our thoughts. However, as Foucault pointed out even deeper than the paradigms are the epistemes, i.e. the fundamental categories of our thought which in philosophy we know as philosophical category theory such as that of Aristotle and Kant. Even deeper is the ontological level of our understanding of the world. In order to ground our design theories it is necessary to articulate each of these deeper levels of understanding. Each level has its own emergent qualities that need to be explored and brought to the surface for our contemplation.

When we look at the level of ontology what is found is that this level has become fragmented. Being itself is a paradox and in order to make that paradox comprehensible by reason we apply Russell's Theory of Logical Types in order to disambiguate it. This produces a set of meta-levels of Being and a set of types called the aspects of Being. The set of meta-levels that are discovered by Continental Philosophy over the last century can be enumerated as Pure Being, Process Being, Hyper Being and Wild Being. The series of types that appear at each of these levels are called the aspects of Being which are Reality (x is), Truth (x is y), Identity (x is x) and Presence (this is x). The aspects are the grammatical uses of Being in our Indo-European languages. The kinds of Being are the levels of intensification as Being folds though itself as it devolves into chaos of ultimate paradox and absurdity. We begin with doxa, opinion, which devolves into paradox which then devolves into vicious circles that again devolves into absurdity and finally ends in insanity, i.e. the utterly irrational. Doxa is the obverse of reason in Plato's divided line. Reason goes through a similar spiral but instead in a different direction in terms of the providing of grounds. There is reason which evolves into self-grounding which evolves into mutual grounding, which evolves into community grounding, which evolves into the supra-rational. In other words when we provide reasons for our actions we normally search around for external grounds beyond ourselves. But eventually it is realized that the best kind of reason, i.e. the most stable kind of reason is that which is self grounding, i.e. appeals only to itself. However, eventually it becomes clear that the self that it appeals to is not unified, as Nietzsche contends. So we see that there is a progressive fragmentation of the self first into something which is the dual of itself and then is multifarious. For instance, in formal systems we know that the axioms form a set and sometimes lend themselves to mutations that produce complementary formal systems that are intrinsically different. These complementary formal systems, such as in Euclidian and non-Euclidian geometry, together give each other a mutual grounding. But eventually we discover that each axiom is itself subject to various interpretations and we need something like Rescher's method in Cognitive Systemization of revisiting the various axioms of our system in turn in a kind of hermeneutic circle in order to successively reground our enterprise. Ultimately we realize that the splits in the self which appeals to itself as a ground produce fundamental discontinuities that are ultimately supra-rational. This supra-rationality is the opposite of the insanity that doxa devolves into. In fact each stage of evolution toward supra-rationality is balanced by the opposite stage of devolution into insanity. The kinds of Being represent the phase transitions between these various levels of devolution and evolution.

When we ask to ground our systems engineering practice what we enter into are these planes of successive evolution of reason and devolution of opinion (doxa). This is what causes the frustration that we have with not finding an easy access to the grounds of our discipline. All this may be summarized by the Idea propounded by Nietzsche which is that ultimately it is groundlessness itself that is our grounds of our discipline. What we are looking at with the successive devolution and evolution of *doxa* and reason is the groundlessness of Being. If we accept this then we can begin to ask our question again, how can we ground our discipline in the groundlessness of Being? Grounding in groundlessness in some way accepts the impossibility of producing firm and incontestable grounds that will never change and accepts that all grounds we might find are temporary and tentative. Ultimately this means that the best we can do is to project Russell's Theory of Higher Logical Types onto the grounds of insanity and supra-rationality in order to disambiguate it in progressive emergent levels. Therefore, seeing the emergent models of the kinds of Being and the aspects of Being embodied before our eyes are basically the best we can achieve. When we see that model we have seen our own grounds to the extent we can have temporary and fragmented grounds.

Now let us return to our concern in this paper with the grounding of Meta-systems theory as the context for Systems theory. What we shall propose is that we use a modified form of Mathematical Model Theory. Mathematical Model theory attempts to work out the relation between Mathematical Categories and First Order Logic. One definition of it is the combination between universal algebra and logic. Now the problem is that all mathematical objects are purely present-at-hand, i.e. they exist only in Pure Being. What we need is something more robust that is articulated at all the various meta-levels of Being so that it is useful in dealing with the real world. Also there is the problem that logic only deals with the values of truth and does not consider the other aspects of Being. So it is clear that what is necessary is something that accepts the fragmentation and ultimately accepts groundlessness by expanding from mathematical model theory into something deeper which takes into account all the aspects and kinds of Being. In this way we will have something robust enough to guide our work of systems design within the context of the real world.

Here we can only sketch out what this meta-model theory might be like. Actually developing it will have to be left to further study and fuller exposition at a later date. Meta-model theory must cover not just the mathematical categories but also the schemas, categories and higher logical types which appear at the successive emergent levels that ground our design theories. Thus we want a considerable expansion of scope beyond the concern of mathematics per se. But also we want to not limit it to first order classical logic. Rather we want to consider deviant logics that comprehend paradox and absurdity as well as supra-rational states as that indicated by the tetralemma (a, ~a, both a and ~a, neither a nor ~a) which considers para-consistency, para-completeness, and para-clarity.

One way to begin is to think of the formal system's properties. They are consistency, completeness and well-formedness (clarity). When we produce a set of requirements or a design we would like it to have these properties. However, we recognize that if even small logical systems are incomplete vis a vis Godel's incompleteness theorem, then our much larger systems will certainly be incomplete as well. However, we would like to strive with our formalisms to define as best we can our systems designs in such a way so that they have these properties. But rather than just ignoring the violations of these properties we need logics that deal with the failures to achieve these ideal properties of formal systems. But beyond this we need logics that will allow us to deal with the real world, i.e. that distinguish values other than just truth such as reality, identity and presence.

In order to set our designs on a formal footing let us adopt for talking purposes a particular formalism that is well suited for use by Systems Engineers for designing systems called Gurevich Abstract State Machine Method. This method was developed by Gurevich to embody Turing Machine descriptions without the cumbersomeness of the Turing machine notation. It has been used successfully to describe all manner of computer languages, and if it can describe the idiosyncrasies of computer languages then it can certainly describe everything that is computable. It is very simply described as a method, one merely describes everything in rules such as one would create for an expert system. The difference is that these rules stand as a static description of the design itself rather than being used as an implementation. It is interesting to note that the rule, i.e. the if…then… statement has an amazing flexibility to describe software systems. In the rule the four viewpoints one would like to represent of agent, function, data and event are unified. What is even more interesting is that we can use these rules to describe systems of constraint on the system or the response of the system itself. Thus the rules may be used to describe either the system or the meta-system and thus may play a pivotal role in our attempt to understand the difference between these two ways of looking at things. Meta-systems are basically filters that operate on systems. Meta-systems are described by a series of niches to which they supply resources for the systems that inhabit those niches. The meta-systems are the origin of the systems that come to inhabit their arena. They provide a boundary within which the systems have free play to the extent they are not confined by meta-system constraints. The meta-system has templates by which it knows how to construct instantiated systems within it boundaries that are the sources of those systems and anti-systems that compete within its environment. A good example of a meta-system is a market where competition between agents occurs within a set of guidelines or rules and given certain limited resources.

Given our ability to define meta-systems and systems with rules that amount to a turning machine representation in the case of the system or universal Turing machine representation in the case of the meta-system, we can go on to look further at our meta-model theory as a means of grounding these representations. The meta-model theory needs to begin with universal algebra and add to that a kind of logic which can comprehend paradox and absurdity as well as taking into account all the aspects of Being. We can begin with the work of NS Hellerstein and his development of <u>Diamond</u> Logic based on the work of G. Spencer-Brown in <u>Laws of Form</u>. Diamond logic looks at truth and falsehood in terms of a dynamic system where these values are repeated. It defines four truth values: ttttt = **True**, ffff = **False**, tftf = **i**, and ftft = **j**. These oscillating truth values are seen as fixed points of paradox. When we combine **i** and **j** by a meta-oscillation between them then we get a vicious circle and when we fuse them then we get absurdity. Diamond Logic comprehends all three levels of the devolution of paradox to vicious circles and absurdity. Even though Hellerstein would like to consider the interpretation of **i** and **j** in terms of both and neither suitable as well, we will reserve this interpretation which gives access to supra-rationality. The fixed points are best interpreted in the way Hellerstien does as *true but false* and *false but true*. Interestingly it does not matter which **i** and **j** are assigned to ultimately the fixed points are indistinguishable except from each other. We may distinguish them if we use complex numbers to do so. In other words if we treat the logical values as if they were numbers we can distinguish the **i** and **j** by treating one as real and the other as imaginary. Their combination is a conjunction of the form ax+bi. Hellerstein himself says that he considers his logic the two dimensional extension of logical values equivalent to the complex numbers. What he misses is the possibility that the logical fixed points may be treated as numbers as well as logical values. In that case we can distinguish them by designating one as a real number and the other as imaginary. Now the change that we would like to make to Diamond logic to convert it into Vajra Logic is to allow all the aspects of Being to become values with respect to the logic. There are in fact four orthogonal values that the logic must deal with which are true/false, real/illusory, present/absent, and identity/difference. These also need to be considered dynamically and each pair in the diachonic logic produces its own fixed points so that ultimately there are eight fixed points rather than just two. For instance, rrrr = Real, uuuu = Unreal or Illusory, ruru = real but illusory = **k**, urur = illusory but real = **l**; iiii = Identity, dddd = Difference, idid = identical but different = **m**, didi = different but identical = **n**; pppp = Present, aaaa = Absent, papa = present but absent = **o**, apap = absent but present = **p**. We would like to suggest that these new fixed points form sets by conjunction with the Diamond logic fixed points. In other words, a Diamond together with one of the other aspects together forms a higher level logic called a Vajra. In that case the fixed points may be treated as a quaternion (x+i=j=k). Vajra's are a kind of sword of discrimination that appears in Buddhist Tantric symbolism. A vajra may be single ended, double ended or crossed with four ends. We see the double Vajra as being the combination of all four aspects into a single higher level logic. In that case the eight logical fixed points (i-j-k-l-m-n-o-p) would be treated as if they were an octonion (x+i+j+k+E+I+J+K). This means that we can have not just logical paradoxes, vicious circles and absurdities but that these may interact with similar conundrums of identity, presence and reality. This can produce very sophisticated combinations of these various forms of higher level paradox, vicious circles and absurdities. This variety of interacting fixed points is exactly what we are confronted with when we attempt to build real systems in the real world. The other three properties that emerge when we add reality to identity-presence-truth of the formal system, are coherence, verifiability, and validity. It is precisely the later that have become so important in Systems Engineering where we attempt to design systems to meet their requirements and to function successfully in a real environment. Within the Vajra logic these properties appear along with the normal properties of consistency, completeness and clarity through the interaction of the various logical values. By treating fixed points as algebraic values we get a complete unification between the universal algebra and logic that is impossible with first order logic alone.

However, it is necessary to recognize that the Vajra logic itself is not merely the combination of four Diamond logics aimed at the different aspects of Being. Rather the Vajra logic has its own emergent properties which can be seen in August Stern's Matrix Logic. It is in Matrix Logic that the tetralemma comes into play giving this logic a supra-rational aspect. Matrix Logic is a combination of Matrix Mathematics and Logic. In it truth vectors are operated on by two by two truth table matrices. Truth vectors may take orthogonal forms of either *bra* or *ket* and these are interpreted as having values of true, false, both or neither in a quite natural way. However, Stern does not interpret the fact that the *bra* and *ket* truth vectors are orthogonal to each other. We can interpret this by saying that these orthogonal vectors are related to different aspects of Being, rather than the same aspect. Thus we could see the matrix logic of Stern as the emergent logic of the relation between aspects of Being. Stern shows how the matrix logic can produce scalar logic values equivalent to the lower level Diamond logic values or if we reverse the operations then we get the production of truth tables. Matrix logic therefore spans the logical levels of scalar, vector and matrix where different complexities of terms appear. Matrix Logic becomes a Vajra logic merely by allowing the various orthogonal vectors to implement different distinctions between the various aspects of Being. As Stern shows this matrix logic which again combines mathematics and logic together allows even for the computation of truth tables alone to produce autopoietic structures. Matrix Logic as an emergent level above even the deviant logics provides a clear picture of the logic of the meta-system. The meta-system is not something necessarily vague and indiscernible. It has indeed its own logic. The problem is that this logic is quite complex in the ways that Stern outlines. As we understand Matrix Logic in the context of all the aspects of Being, as a Vajra Logic then we will come to get a very precise picture of the operation of the Meta-system. The way that the Matrix Logic introduces orthogonality and also the relations between various values of the *aspect*, *non-aspect*, *both aspect and non-aspect* and *neither aspect nor non-aspect* is the means by which supra-rationality may be seen to enter into picture to balance paradoxicality, vicious circles and absurdity.

Meta-model theory when taken in relation to the Vajra Logic gives us a basis on which to ground our design of real systems. Not merely producing formalisms that are divorced from the real world, but ones that deal with reality as an independent aspect orthogonal to truth, as well as identity as orthogonal to presence. When we combine this with the ability of the Gurvich Abstrace State Machine to model Turing and Universal Turing machines and thus both systems and meta-systems then we suddenly have a profoundly robust, not formalism, but systemism and archonism, where an archon is another name of a meta-system. When we produce our rules in such a way that they are articulated in terms of not just truth and falsehood, but in terms of reality as well perhaps in terms of succeed and fail as we see in the SNOBOL, ICON, UNICON languages, then we will be able to model in a very sophisticated way the kinds of situations that obtain when we interface a system to the meta-system of its ecology or environment.

Model theory helps us because it takes an arbitrary language is its source for producing the model of a mathematical category by assigning values of true and false. What we want to do instead in Meta-model Theory is to produce languages with sentences that we assign values of not just truth but also reality, presence and identity in order to describe meta-models not merely of mathematical categories, but also of schemas that are the core of systems designs which are inwardly dependent in turn on philosophical categories and ontologies. These meta-models can be considered in turn in terms of the deviant logical forms that appear with the Diamond, Matrix and Vajra Logics in order to understand more precisely the nature of the diachronic meta-models that found our formalism. A formalism for such languages has already been presented in the work Wild Software Meta-systems in which the ***Integral Software Engineering Design Methodology*** was formulated. This methodology assumes that there are four fundamental viewpoints on any real-time software system which are Agent, Data, Function and Event. Each viewpoint interacts with the other viewpoints through a bridging methodology and for each methodology a minimal language is produced. These languages are more expressive than current graphically oriented design languages. The combination of the languages that describe the minimal methods for real-time software design allows us to construct a meta-model of the system under design. It is right to call it a meta-model because it is comprised of various models that are grounded in the various minimal methods that arise from the interaction between viewpoints. We need only raise these models and apply them to a higher level of abstraction in order to make these methods applicable to the entire system rather than merely the real-time software element of a system. The meta-models of the designed system are described by sentences composed out of the minimal method languages. However, what we see is that the difference between the syntactic level where consistency, completeness and clarity operate is complemented by the semantic level where validity, verifiability and coherence operate. What is interesting in this respect is that signification appears by the addition of the aspect of reality to the mix. In other words a formal system already encompasses identity as tautology and presence as the existential instantiation of variables. What is lacking is the distinction of reality. When reality is added then the semantic level is achieved where signification is produced. So at the heart of model theory is the basis for the creation of meta-model theory which then can be expanded to describe schemas, categories and ontological commitments.

What we find is that requirements once aphoristically stated can be converted into a Gurevich Abstract State Machine formulation that is a concrete interpretation of those requirements. In this representation there are myriad rules that all embody the fusion of the data, function, agent and event viewpoints. But when we move to design then we use the languages of the minimal methods in order to describe the various meta-models encompassed by our design. Here the viewpoints are separated and their interactions specified via their interactions through the minimal methods. But the minimal methods together by giving us slices of a Turing machine allow the computation to be further specified. This specification of the design is then implemented with a programming language. For prototyping we might use a very high level languages such as UNICON or other lower level languages such as C, C#, C++, Java, Eiffel, etc.

But we must remember that all these various transformations of the meta-model are still determinate. In order to produce a more robust modeling capability we must remember the other meta-levels of Being and their mathematical concomitants. Pure Being is represented by the Calculus, Process Being by Probabilities, Hyper Being by Proclivities in the form of Fuzzy or Rough Math and Logic, Wild being by the propensities that we see in Chaos Theory, Fractals and Vagueness. We must be willing to increase the range of our models by adding these various forms of mathematics as a means of coming to terms with the relation between our world and our designs of the things that are expected to fit into our world.

But also there is a concern that our designs now consider the diabolical use of our own technological infrastructure against us. This makes the drive to understand beyond systems to meta-systems beyond formalisms to deviant logics more pressing. As explained in the paper "Anti-Terror Meta-systems Engineering" the wider view of nested emergent schema can help us look for those gaps and blindspots that an enemy might exploit. It calls us to develop twenty-first century systems theory and systems engineering today, by recognizing how they are expanded to include meta-systems theory and meta-systems engineering as well as other schemas that fit within our philosophical categories and express our ontological commitments. This paper sought to bring some clarity to the relation of meta-mathematical meta-models and Vajra logics to this enterprise. Hopefully with these sophisticated tools we will be able to head off disaster before it happens as well as make our own systems more safe and secure as well as robust across many different qualities beyond those. Safety or security are properties of systems that need to be added to those that occur naturally from the interaction between the aspects of Being. The six fundamental properties are consistency, completeness, clarity, coherence, verifiability, validity. If we want to describe other properties such as security and safety we need to add sets of rules to our meta-models that distinguish those properties. But those kinds of properties call for an understanding of failure, failure to be safe and failure to be secure. Those failures occur because the meta-system is more complex than the systems that we build to inhabit them. Thus, our logics need to be robust enough to handle not just paradox, vicious circles and absurdity with respect to truth but also with respect to reality. It is those conundrums that we are designing against that need to be explicitly modeled and we need a logic like the Vajra Logic built upon the foundation of Matrix Logic to accomplish that. It is a dangerous world out there which goes beyond our assumptions in ways that are difficult to anticipate. We need to arm ourselves against that world with a kind of meta-model theory that includes deviant logics and goes beyond mathematics to consider the basic structures of our design theories such as schema theory and philosophical category theory. We are continually applying these structures of projection in our work as systems engineers. To the extent we can make them more prominent and conscious the more we will reduce our blindspots and will thus make ourselves less vulnerable to attack through the gaps in our own self understanding of the technological infrastructure that we ourselves produce.

This brings us back to the question of grounding. In our designs we appeal to multiple reasons as a basis for our design actions. But one things we need to understand is how much the design activity is self-grounding, i.e. self-fulfilling. When we design we continually revisit the axioms of our requirements. Many of these are mutually grounding or even grounding as a community of axioms that we treat with a kind of Cognitive Systemization described by Rescher. But ultimately the discontinuities between the axioms remain as a supra-rational ground. However, what we do not do is look at the requirements of the meta-system, the domain, and the world. This broader horizon of requirements needs to be taken into account in order to provide the basis of designing the meta-system, domain and world that the formal structural system is to be embedded in. These broader environments are not just systems again but something very different, in the way an operating system is different from the applications that it encompasses. The broader environments have different kinds of requirements that have to do with the interoperability of the various technological systems that form part of the technological infra-structure. When we turn to these requirements and realize that they appear in a what Bataille calls *General Economy* rather than an ordered logical and rational *restricted economy*, then the real need for meta-models and deviant logics becomes clear. This is the horizon of exploration for a twenty-first century Schema Theory and Schema Engineering which will replace what we now call Systems Theory and Systems Engineering. It is the hazards we have found in the world that drive us toward the exploration of this horizon where meta-systemic environments, domains and worlds need to be designed just as much as the systems we have learned to design in the last couple of centuries. Twenty-first century systems engineering will be much more complex and sophisticated than anything we have put into practice up to this point. But we must rise to the challenge in order to advance from systems design, to environmental meta-systems design, to cross-environmental domain design on to the design of future worlds.