4

SELF-ORGANIZING VIRTUAL ENTERPRISES

H T Goranson
Sirius-Beta
1976 Munden Point Road
Virginia Beach, Virginia, 23457-1227, USA
tedg@sirius-beta.com

We reintroduce a model of virtual enterprises that is based on features, from customer needs to delivered product. The features in this case correspond to agents that exhibit self-organizing, emergent behavior. The most innovative notion is that features have all the mechanisms necessary to evolve an optimized virtual enterprise. The notion of "narrative urge" is introduced.

The first part is some history of the discovery of types of virtual enterprises, placing the reintroduced concept in perspective. Then we define the notion of a functional agent, feature-driven domain intuitively, indicating some challenges.

Then follows an overview of a general theory of emergence. The virtual enterprise case is mapped to that. A scenario is indicated. Due to space limits, only an overview is given.

1. INTRODUCTION

Two decades ago, the US military performed some large classified studies of the state of the manufacturing enterprise. The impetus was a by now well known problem: doing complex work in large enterprises is so heavily burdened by the machinery of running the enterprise that things cost too much (typically twice), they evolve too slowly in terms of technology and some things just cannot be done at all.

Three radically different models of the future enterprise emerged. One seemed most natural to power brokers at the time, the "integrated enterprise." In this model, large relatively stable enterprises became more "frictionless" with information flowing along designated lines of command and control. The enterprise can be and often is "virtual" in the sense that many companies are "linked" in a supply chain. A characteristic is that the integrating strategy and management is integrated with the management of production and capital at the top, with a prime contractor or leader.

A second vision was rather radical. It attempted to decouple the three (production, capital, integration). The integrating technologies and methods would be provided from the surrounding environment, perhaps as standards, best practices and distributed facilitating agents. This was termed the "virtual enterprise," with a trivial version consisting of prequalified partners that could easily "plug and play." In this vision, the prequalification would be performed by a sort of virtual prime.

A more advanced virtual enterprise notion depended on more flexible, federating technologies and methods, allowing companies that had never met to aggregate to perform work that they may have never done before. Still, in this vision, one needed agents to support some management of production, capital and integration. The goal was that these three functions need not be linked, but in practice so far, they always have been and performed by a partner in the virtual enterprise.

US research became overtaken by political forces, and the sponsorship there stuck at integrating technologies to help (politically powerful) large enterprises become more so. Research into the virtual enterprise case has been to a degree carried by the European Community, but has similarly stuck on the trivial case of relatively small pools of pre-integrated potential partners.

It is unlikely that either of those business models will advance much beyond the rut that they are in, technology not-withstanding. There just is not the government will, and the academic research communities have dug into existing power centers, an unfortunate byproduct of the way research is funded, dominantly through political and market forces.

It is time, in our opinion, to revisit the third business model, the so-called "self-organizing virtual enterprise." The reason for trying to run before we walk should be obvious. If we can propose a model — however radical — that has sufficient financial benefits, perhaps a kind of discontinuous, beneficial change can occur, the type of change that characterizes the best of technology and provides the stuff for societal optimists.

In simple form this model considers all the components of the potential enterprise as active agents. The level of granularity is presumably fine, so we are not talking about companies, but small workgroups or even individuals within them. Each of these has some sort of independent agency.

This model is not so hard to envision. The system would be triggered by some enterprise goal, presumably a product design and the many, many agents would go to work, negotiating with and informing each other to produce and continue to refine the virtual enterprise. The research challenge here is how to manage multilevel optimization.

Agent systems usually work best when one goal is desired and that goal provides a consistent narrative. The problem is that product designs have little to directly indicate enterprise goals (like profitability and stock value), so need to be translated. Also, there need to be selfish organizing imperatives at many levels, the agent (whatever the granularity), and on up through several layers including the containing small firms then further up to the overarching enterprise.

At the same time, there are "vertical layers," the enterprise-wide functional flows of quality and financial measurement and control, for instance. There seem to be eleven of these, each of which has its own emergent behaviors to optimize, as different in ontology as many of the horizontal organizational layers.

We've made detailed examination of the technical problems associated with this multi-level emergent behavior when you are presented with product models or features. There are profound difficulties with this because there is no way at present to formally harmonize the parallel emergence. A tentative consensus is that until we have a better way of managing "semantic distance" as a federation metric, this vision will not be supported.

We believe that a better approach is to look at the more "advanced" concept of self-organizing virtual enterprises. This vision of the enterprise is much harder to describe. (There are some in the business world who just cannot "get" it with any effort.) Though this enterprise notion is more arcane in some ways, the technical problems in emergent systems design become more tenable and possibly completely solvable with existing conceptual tools and methods.

2. A MODEL OF EMERGENT BEHAVIOR

2.1 Features

In management schools, you'll often hear talk of a customer-centric enterprise or "value chain," one in which there is tight coupling to the customer's needs. If it were a virtual enterprise, the customer might be considered one of the partners.

If the old model is that the customer is there to sell stuff to so as to benefit the enterprise, a new model might be that the enterprise is there to benefit the customer. In the former, the customer incidentally benefits (or is convinced so) while in the latter it is the enterprise that incidentally benefits.

Our model, the one we explore in this paper, differs from either of these. It is a matter of who has the primary agency. The first, normal model is that the enterprise forms around a product. The enterprise components have the agency. In the second model, the customer enters the virtual enterprise and serves many of the roles of the prime contractor, the organizing agent.

What if the product model itself (instead of any group of people) had the agency? But a product is just one instance of a collection of customer needs instanced according to certain constraints. What if the features of the model themselves had the agency? That's precisely what we have been investigating and report on here.

In fact, we'll go a bit further toward the unusual but workable — quite a bit further — and apply some newly maturing ideas on emergent systems theory.

The basic idea here is to model the enterprise not in terms of entities that have functions and take actions, which is the usual way. Instead, to model the world of the enterprise as consisting of "urges" or desires. Perhaps even the term "seductions" could work. In this model, the elements with agency are the urges themselves and they assemble entities into virtual enterprises as a way of expressing those selves.

In the conventional model, components of the virtual enterprise exhibit emergent behavior (that is, they have agency) and the product model (with other models like metrics of success) provides the map, concept or "story" around which these elements form structure. In the new model, the elements of the product model are a collection of urges that have agency and they arrange the elements of the virtual enterprise.

There are precedents for this sort of shifting of the frame of reference. The most familiar to engineers will be shifting from the time to the frequency domain. In software, there is a better known and more widely debated schism between those that model code procedurally (like Java) and those that break the problem down as functions that transform, so-called "functional programming."

Other metaphors concern the shift from a noun-oriented breakdown to one based on verbs. Also of interest is the change in genetics from thinking about genes as a collection of molecules that act, to a model of information packets that "act selfishly" and in the process incidentally create their own type of enterprises: humans and societies. In logic, the analogy is to shift between a set-theoretic foundation and one based on category theory, the theory of functions.

As it happens, all of these: logical, genetic, programming, linguistic... they together provide a coherent, rigorous theoretical basis for supporting our new way of looking at emergent behavior of all kinds. We appropriate that in our model of emergent behavior in virtual enterprises.

(It should be remarked that this notion is quite different than incubated in the "complexity theory" domain. The difference is one of whether the semantics are geometric or algebraic, a discussion of which is beyond this paper.)

3. THE FEATURE-BASED ENTERPRISE

3.1 General

Without the notions of agency and emergent behavior, the idea of looking at features is very intuitive. Features in this context can easily be seen the way they commonly are in the Computer Aided Design modeling world: the constituents of a product model.

It is easy to think of a model of customer needs that transmutes to a product model, or many successors, each of which lead in concert to a production and management model with thousands of associated process models. And it is relatively easy to think of these as composed of features which collaboratively deal with each other, transforming each other, forming societies from which the next level emerges and shaping elements of its surrounding to incidentally produce a healthy, profitable virtual enterprise.

Seen this way, it is not so unintuitive to think of the world in terms of selfish features, each of which has agency, all of which can be modeled functionally. These functional models are more than ordinary models that represent; they can be executable code that can simulate or control the emergent behavior.

So, our model of the self-organizing (virtual) enterprise is based on the notion of seeing the world as a collection of urges, expressed as features that have agency and act as functions, exhibiting emergent behavior.

3.2 The "Layer" Problem

We still have the layer problem, but it is more tenable.

The layer problem is simple to describe. In the general emergent systems world, it is often explained thus: we have a good theory of chemical interactions and we have another quite different theory of biological interactions. Each is in their own "layer." But clearly there are actions in one layer that affect behavior in the others, and in fundamental ways. How does the language that molecules "speak" to one another relate to the different language cells use?

Now shift the problem into a more difficult context by introducing the self-organizing notion. Chemical elements in this world clearly self-organize with one result that cells come into being. And throughout the life of the cell there is information conveyed "up" and down that layer boundary, non-trivial information related to organizational urges and scripts.

The similarity to the virtual enterprise case is obvious. If not, the reader needs to be reminded that individual reward systems are radically different than those of work cells and plants and small companies, on up to the enterprise and beyond to healthy societies. The ontological differences in these layers are of the same significant order in the enterprise case as in the layers of physics, chemistry, biology, organisms and societies. (The top layer: “societies” is the same in both cases, virtual enterprise and general science.)

This layer problem is well known, an open problem in science. Interim solutions involve either transporting a mechanism from the top or the bottom. The "bottom" here means physics and the concept is entropy, the degree of organization of a physical system, characterized statistically. Clever thinkers have figured out jury-rigs to apply a "negentropy" principle to organizing tendencies at higher levels, on up to economics.

Transporting down from the top is equally popular. A prevailing theory of information and language in humans is semiotics. Similarly clever thinkers have applied this very human-oriented principle of representative meaning to the lower levels and "biosemiotics" is an active field. Both of these approaches work well enough in limited cases to survive. But both lack the formal depth a solid solution to this layer problem needs.

What makes the layer problem so much easier now that we move to a categoric feature space for the enterprise is that we build an equivalence between features and transforms in the form of functional agents. Features in one layer can act on each other or aggregate with each other to form features of completely different types (according to strict rules) at higher or lower levels. The problem of transforming and shifting levels of abstraction are "built in" to the space, as it were. Our only problem is in defining the functional transforms. We approach this below by ordinary group operators.

3.3 The Problem of "Scripts"

All of the familiar enterprise models have agents and scripts. We’ve made the agent problem go away, but what about the script problem? By script, we mean the rules by which an enterprise is organized. Each agent in the ordinary models has access to its part of the script, even if it that script is somehow built in. To simplify this in the enterprise case, these scripts are based on vertical domains with relatively orthogonal, stable and mature concepts. Thus, we have financial infrastructure, product design, sales, human resources and so on, each with a set of rules, abstractions, metrics and information flows.

If we do away with agents reading scripts, we need to be able to have our functions reinvent them on the fly them to a meaningful degree. After all, what kills enterprise engineering the way we do it now is that we have to figure out the scripts (process plans, best management practices, performance metrics and so on) and maintain them in parallel with building and maintaining the enterprise. The main advantage of self-organizing systems is that they should be able to figure most of that out without human guidance — and possibly do it more cleverly.

So we have the script problem. We had it before, partially managed by cleanly dividing the enterprise and separating it, only to have to integrate the pieces. Now we have it in a different, possibly more tenable way.

What makes it more tenable is that instead of dealing with constraints and normal forms, we can reinvent approaches on the fly based on goals. And isn't that the point of self-organization, to come up with optimized structures that work in unfamiliar ways that we never would have thought of?

Naturally we still need all sorts of constraints, legal, ethical and so on, to have as external references, boundary conditions.

3.4 The Problem of "Memory"

The final problem is termed "memory" which we will show below is a key concept. But the problem definition is larger, one of introspection of the enterprise, pattern matching to prior situations and prediction based on history.

This problem comes from the plain fact that we can reinvent how enterprises form and operate, but we cannot reinvent what surrounds them. Among other things that will remain (like customer infrastructure and legal constraints), we have the problem of finance. Finance is based on explanations and predictions and despite the common use of the term "management science," it is an inductive science where repeatability is the rule.

You cannot understand something in business unless you have seen something like it before. And if you cannot understand it, you cannot finance it. A huge collection of monitoring and reporting tasks usually support this notion in the ordinary enterprise. But if we have no scripts and we don't even know who the actors are until they announce themselves — if we have no idea what the enterprise will look like or even what it will make, how can we fuel it with capital?

In theory, this new model takes us much closer to a market force driven economy, but in some respects further away from capitalism. To mitigate this, we have to introduce "apparent determinism." The self-organizing system doesn't have to be deterministic, it just has to look that way to the financial linkages from the outside.

The problem of memory and scripts we solve with a system of concepts, formal and intuitively informal, that we term "narrative." Each of our elements is a feature based on an urge. Each urge is a tendency to want to make a story. Different stories compete, some fitting the constraints better than others, some singing more eloquently.

Formally, each feature is a function with transformative possibilities toward certain complex situations, the situations specified by a situation logic. These "situations" capture the "what" we'll call narrative. but we get ahead of ourselves.

4. A THEORY OF EMERGENT BEHAVIOR

4.1 General

In this section, we provide some overview of work being done in new theoretical foundations for emergent systems. The idea is to apply this larger science to the virtual enterprise domain we've been discussing.

As noted, the problem of a coherent theory of emergent behavior is a vexing one. A few pockets of suboptimal solutions exist. We do not explore here why complexity theory, biosemiotics and quantum statistics are inadequate for a general theory of emergence.

Our requirements are for a system that can both describe how a system works and be used by that system internally. In other words, it should work if you are inside and part of the system and/or outside the system watching or perhaps engineering it.

The domains of application will include human systems (like virtual enterprises) and non-human (like molecules and the features or urges we've noted). The types of system will be mixed between natural systems (like the behavior of molecules) and engineered (like the behavior of programs that schedule work according to business rules).

The idea is not to just describe but to create and not in any one domain but in mixed domains. The special case is where one domain like an enterprise or human body can emerge from components of lower levels like features/urges and molecules.

We've found three principles that seem promising, both in the case of general emergent systems and the self-organizing virtual enterprise as we've described. Each has formal mechanics and a human-specific metaphor.

4.2 Identity

Instead of particles, fields and forces, we look at things in terms of urges and narrative fragments. Urges have and generate associated particles in a complement to the generally accepted notion of particles and their associated fields.

From the normal representations we abstract three new functions associated with identity, introspection and equivalence. These can be seen as urges themselves and have identity as functions in their own spaces and types in each other's space. Categories collect each type and relationships among them and functors build groups used below,

Essentially what's happening is that we have an ordered space to talk about what something is, what it wants to be and how it "thinks" about that. This ordered space also contains the relationships among these abstractions as first class abstractions in the space. The point is to give us a well ordered concept space so that we can reason among and operate on elements in the space; where all the abstractions have a metaphoric equivalent, namely "urge" toward "narrative," as intuitive and no more arbitrary than "particle" and "field;" and where the relationships among the different citizens are functors.

That's the most esoteric part, creating the abstractions. The intuitive metaphor of narrative is easy, but settling on the categories needs specific mapping from each domain. We expect associates to assist with the hardest domains: physics and chemistry. The enterprise case is tricky as well. Actually it is easy in a generic, test case where we can arbitrarily model processes using anything that works.

But real enterprises have what we've called vertical layers, independent domains that do accounting and management of different types. Human resources and production for instance, also finance and strategic planning. Industries have their own peculiar frameworks in this regard. Doing the categoric mapping is relatively easy to do but requires lengthy refinement and validation to be trustworthy for the "apparent determinism" we mentioned.

This notion of throwing everything into category space is common in functional programming and follows a suggestion made by Saunders Mac Lane and fleshed out by Jon Barwise.

4.3 Aggregation and Transformation

Where the first set of formalisms deal with how the representations of the concept space are determined, this second set deal with how they interact. It is the rough equivalent to the operation of the enterprise. It is how things interact with each other, exhibiting emergent behavior that structures systems and operate those systems.

In this, we follow the dominant tendency and employ group theory. It will appear that we are different than most applications because the abstractions are a bit unfamiliar, but we simply apply the wreath product to two interacting collections to discover potential higher level groups derived from, but still linked.

It is the formalism we use to link layers and was inspired by mechanisms described by Michael Leyton that manage the two most important characteristics: "precedence" and "history." We have to explicitly provide for these because normal notions of time and causality are lost in our new identity domain. That domain is inherently stateless, so memory is captured as a new, persistent, higher level.

The group operators predict and generate the new layers.

4.4 Logic

We are indebted to Jon Barwise and others for this component of the approach as well.

We need a logic to apply in reasoning about all this. It has to be "soft" and deeply introspective for the humans reasoning about everything, including the above. It has to have a different sort of softness for humans reasoning about systems they are parts of.

The soft (meaning unknown or unrepresentable) facts in this new logic are explicitly captured as "situations" or components of situations, and the extended logic is known as "situation theory." There is a corresponding situation semantics and logic. It is hairy compared to good old first order logic, and costly to employ, but the world is a hairy place.

We've adapted it slightly to accommodate the soft urges that non-human speech acts would convey or follow in the urge-based agents.

At the time of his death, Barwise was harmonizing the first and last of these theories (category and situation theories) in something he called "channel theory" as a basis for a general theory of information flow.

To summarize: we represent the world in terms of verbs or functions and to do so, employ category theory to make our concept space well ordered, better ordered than what we inherited as an accident of history. This is truly a world model, a general abstract semantics for every domain and tendency. The metaphor is urges toward global, multilevel (or folded) narrative.

Once that is done, the functions apply to each other in the normal way, some of which transform elements from one domain of reality to another, "higher" one. The topological transforms of group theory are used in those functions, with different topologies corresponding to different functional groups.

That's the internal machinery of how the functions trigger and cluster. But we need a logic for elements to reason about other elements. It has to be more flexible and accommodating than first order logic, so we employ a slightly extended situation semantics.

4.5 The Virtual Enterprise Domain

Our group got into looking at the general problem of emergent systems because of the self-organizing virtual enterprise case. Now we are working primarily on that general case and mapping back to the virtual enterprise domain (and one other).

The virtual enterprise case is more difficult in that at least nominally it is 100% a human system. Although the laws of natural science constrain processes, they are all in the context of human goals. What makes the virtual enterprise case simpler is that we know those goals; they are explicitly expressed and generally simple. Moreover, we can assume that though there are many dimensions, all the players get rewarded in much the same way, measured by similar metrics.

The case in chemistry, for instance, is different. It makes little sense to infer how molecules "think" about rewards. It is quite a bit easier to infer how urges function at that level because anticipation is built into the metaphor. But the notion of successful organizations is still elusive, say for biochemical systems, unless you have a desired outcome in mind.

This idea of engineered systems sits a bit uneasily in the container of self-organizing ones unless we limit ourselves to engineering by setting boundary conditions only. So that is what we must do for systems in which we cannot directly participate, like chemical ones. Human systems differ in that reward systems and new metrics can appear as part of the innate tendencies in the system. In this way the virtual enterprise case is simpler.

In our experience, the business case is much more difficult in getting the basic abstractions right. As noted, the layers aren't as coherent as one finds in chemical systems. They have all these parallel infrastructures, each of which has its own ontology and requires a unique mapping. And different industries have their own semantic peculiarities as well.

To mitigate this effect, we expect to rely on a single normal set of semantics and federate what we find using the formal ontological methods of the Process Specification Language, which we believe can be used to federate function semantics from one functional expression to another. (But this does not help with the problem we encounter when the stuff of representations is incorrect as we find sometimes, or deliberately cloaked, as we often find in criminal or military enterprises.)

4.6 A Scenario

We are working with four test scenarios. Three are only of passing interest here.

One involves a study of narrative itself and specifically introspective, folded narrative as it moves through the popular culture, meme-like and exhibits in film. Another concerns the detailed mechanisms of stem cells and the biochemical urges involved in branching. A third is for self-organized concept mining in large data libraries.

Our virtual enterprise scenario is a refinement of one we have been using for a few years now in workshops.

The customer base in this is initially civilian war injured who lack a limb, a depressingly large group of concern to all in the developed world. Prosthetic limbs are notoriously unfriendly to mass production; each would ideally be fitted or even engineered and manufactured in lots of one.

We have allowed this test scenario to be more complex than supposed in as many dimensions as we could envision. For instance, we posit a virtual enterprise of virtual enterprises, some forming in villages in the developing world, some forming around innovative design resources and non-governmental organizations (which themselves in this scenario might be virtual), and others from components of large established enterprises in the west, all linked in dynamic constellations, emerging and fading.

Moreover, we assume that the features would not only reach as deeply as the design and fitting of personal prosthetics, but their actual composition and control systems as well. That means that some of the features may be creating software or polymers in a self-organizing way within their respective domains. We do limit the emergent behavior to devices rather than services because we want to understand how project model features relate to the feature clouds involved in such a manifold system. Product features are a well understood beast and a necessary link to the apparent determinism strategy used. Ultimately, we wish to extend this to agile self-organizing drug design and manufacture, which is why we are looking at stem cell mechanisms.

5. RESEARCH CHALLENGES

This paper is a research note on a new project rather than the usual report on research completed. It is too early to provide exportable details from the work here. But we do hope to indicate some research challenges.

Concerning the definition of active features: as it happens, the Computer Aided Design community is well suited to address this issue, with research focused on extensions to ISO standard 10303 and existing tools that express similar features. Alas, the process modeling and enterprise integration community has really slipped a decade behind in this. There is no serious effort I know to look at process features outside of the interoperability community.

We need to address this inadequacy fast, as the product data management community is growing in power at the expense of the traditional enterprise integration base. We believe this is why.

ISO 18629 is the Process Specification Language that is the most robust formal ontological approach for general semantic federation of the type needed to federate across domains in the enterprise and among different enterprises. Little work has been accomplished toward this use.

There has been an unfortunate confluence of process modeling for the enterprise and modeling for software, with the unhappy result that research in enterprise dynamics is nearly all object oriented. Yet functional programming is growing as its comparative successes pile up. We need a robust basis in functional thinking in the enterprise, and pedagogical examples in the programming language Haskell.

Though entirely appropriate for this type of reasoning whether function-centric or not, situation theory is cumbersome and difficult to work with for non-specialists. We need a "situation theory light" for relatively unsophisticated users.

The biomedical community has some familiarity with these methods. A better relationship between the pharmaceutical enterprise and the virtual enterprise community would benefit both.

Studies in folded narrative need to be more robust.

6. REFERENCES

1. Barwise J, “Constraints, Channels and the Flow of Information”. In Situation Theory and its Applications, Aczel P, Isreal D, Katagiri Y, Peters S eds. Palo Alto: Stanford University, 1990.

2. Collier J, “Order from Rhythmic Entrainment and the Origins of Levels through Dissipation”. in Symmetry: Culture and Science. 1998; 9:2: 165-178.

3. Gass WH, Finding a Form. Ithica: Cornell University Press, 1996.

4. Goranson HT, The Agile Virtual Enterprise. Westport CT: Quorum Books, 1999.

5. Goranson HT, Cost Analyses and Survey of Meta=abstraction for Modeling. Norfolk: Classified DOD report, 2002.

6. Goranson HT, Chu B, Gruninger M, Ivezic N, Kulvantanyou S, Larbrou Y, Masouka R, Peng Y, Sheth A, Shorter D, “Ontologies as a New Cost Factor in Enterprise Integration“. In Enterprise Inter and Intra 7. Organizational Integration, Kosanke K, Jochem R, Nell JG, Bas AO, eds. Berlin: Kluwer, 2003.

8. Hilbert D, Cohn-Vossen S, Geometry and the Imagination. New York: Chelsea, 1952.

9. Hilfinger PN, Abstraction Mechanisms and Language Design, Cambridge MA: MIT Press, 1982.

10. Leyton M, Symmetry Causality Mind. Cambridge MA: MIT Press, 1992.

11. Kauffmann S, Distinguishing Features. Baltimore: Johns Hopkins University Press, 1994.

12. MacLaine S, Categories for the Working Mathematician. New York: Springer-Verlag, 1971

13. Myers K, Konolige K, “Reasoning with Analogical Representations”. In Diagrammatic Reasoning, 14. Glasgow J, Narayanan NH, Chandrasekaran B eds. Cambridge MA: MIT Press, 1995.

15. National Research Council, Visionary Manufacturing Challenges. Washington: National Academy Press, 1998.

16. Rosen R, “On Psychomimesis”. In Essays on Life Itself. New York: Comulbia University Press, 2000.

17. Stilman B, Linguistic Geometry. Boston: Kluwer, 2002.
18. Vernadat FB, Enterprise Modeling and Integration. New York: Chapman & Hall, 1996.