‘Organizing’, ‘Innovating’ and ‘Managing’ in Complexity Space

Mihnea Moldoveanu
Professor of Economic Analysis and Desautels Professor of Integrative Thinking
Vice-Dean, Learning, Innovation and Executive Programs
Director, Mind Brain Behavior Institute
Director, Desautels Institute for the Future of Learning
Rotman School of Management, University of Toronto
Visiting Professor
Harvard University Graduate School of Business Administration

Working Paper 19-128

Abstract

We two-dimensional measure of organizational complexity that distinguishes between the informational and computational dimensions of complexity and aims to function as a maximally context-invariant environment for posing fundamental questions about organizational dynamics, akin to the role and function of a *phase space* in in classical mechanics. The new measure and associated space allows us to understand the effects of a researcher's or manager's *choice of model or representation* on the resulting complexity measure of a phenomenon and to measure the complexity of any organizational phenomenon that can be represented in a form amenable to an algorithmic description. We use the new complexity measure to present a unified treatment of complexity coping mechanisms, complexity-driven organizational failure and complexity-adaptive innovation and draw new distinctions enabling new questions whose answers will be useful to researchers and executives alike.

Acknowledgments. I acknowledge many useful discussions relevant to the subject of this paper with - and helpful suggestions from – Robert Bauer, Adam Brandenburger, Simon Levin, Nitin Nohria, Martin Reeves and Nassim Taleb.

1. Introduction.

Science is (all and only) that which can be explained to a computer. The rest is art.

Donald Knuth

'Complexity' is a label given by many C-level executives to a wide range of phenomena they report themselves to find 'troubling', 'challenging', 'intractable' or 'unmanageable' [IBM Report, 2013]. But the ambiguity in the referent of the word in business and consulting parlance is matched by the ambiguity of its referent in discourse spanning the natural and social sciences. We most frequently do not know what others think we mean – or indeed what we mean - when we talk about complexity, on account of many, parallel, inchoate intuitions and measures of complexity that are commonly and frequently held, often at the same time, by many people. The ambiguity can serve useful dialogical purposes: It allows conversations between speakers that are in fact oblivious to the depths of their miscommunication to take place in ways that safeguard the illusion of understanding. But it undermines attempts to build descriptive, prescriptive or ascriptive [Moldoveanu, 2011] models that allow us to represent and *intervene in* organizations on the basis of a measure of 'complexity' as a dependent or independent variable. The problem has both epistemological and pragmatic projections. Both need to be addressed to make talking about complexity minimally useful. That is what we intend to do *infra.*

Ambiguity about Complexity: What could 'Organizational Complexity' Mean? The study of organizational 'complexity' is beset by a duality of the meaning of the word *complexity* that refracted onto common language use. The Oxford English Dictionary tells us that two ideas are incorporated in the word's reference:

1. In one sense, an entity is complex if is made up of several interconnected elements that jointly constitute a whole (e.g. a 'building complex'). A complex situation, problem, product, process, procedure or predicament is complex in virtue of its being made up of many simple but structurally or functionally interconnected or interdependent entities and thus 'complex' in virtue of its structure, or dynamics, or of both.

2. In another sense, an entity is complex if it is not simple to understand (in the sense of *Verstehen*), predict or explain. In contrast to 'complex' as an attribute of the components that constitute an entity and of the relations between them,

the second meaning of 'complex' in lay usage refers to the relationship between the 'complex' entity and its observer. This sense of 'complex' matches the intuition that what one observer (at one point in time) finds 'complex' may not qualify as 'complex' for another observer (or the same observer at a different time).

Multiple Referents of 'Complexity'.

'Complex' has been canonically used as an adjective to describe organizations and organizational phenomena [see, for instance, Cyert and March, 1963; March and Simon, 1958; Weick,1995]. It gained prominence in approaches that link organizational design (i.e. organizational structure and processes) to the complexity of the organizational environment [e.g. Mintzberg 1979]. Although less precise than the meaning of 'complex' in the natural sciences, the 'complexity' of the organizational environment was understood as a result of factors such as the number of sources of influence in the environment, the speed and relative degree of synchronicity with which causes and effects propagate through a network of environmental influences, or the sensitivity of the overall environment to single point events or the influence of single social or economic actors [Scott, 1981]. These original intuitions about organizational complexity have been formalized with increasing exactitude by scholars of organizational phenomena [McKelvey, 1999; Cohen, 1999] using structural models developed in the natural sciences (NK(C) systems, for instance). Of course, 'complexity' has many more possible referents than those that have been adduced as dependent variables in well-cited studies. A *person* is neurologically complex, psychologically complex, spiritually complex, morally complex, socially complex – and each use of the word 'complex', supra, will have very different formal representations. Measuring the complexity of neuronal dynamics entails very different a priori commitments than does measuring the complexity of one's moral reasoning patterns.

In the natural sciences, we similarly find many ways of giving meaning to the word 'complex'. Algorithmic information theorists [Li and Vitanyi, 1993] equate complexity of an object with the length of the shortest string of bits that can be used to represent it. In physics and engineering [Casti, 1991; see Shannon and Weaver, 1949] one speaks of the *entropy* of an entity, or the number of mutually exclusive and collectively exhaustive states that the entity can assume in virtue of its constitutive elements and their properties. In theoretical computer science [Cormen, Leiserson, and Rivest, 1993], complexity relates to the difficulty – gauged by the number of required operations - of *executing* an algorithm whose output simulates the behavior of the entity in question, and is referred to as computational complexity. In the general systems theory of Herbert Simon [1962] (which has found resonance in the study of organizations [Miller, 1993; Anderson, 1999]) 'complexity' relates to the degree of coupling of a multi-component system, which is similar to the use of 'complex' in attempts to model biological systems using Boolean networks [Kauffmann, 1993]. But the natural sciences, unlike organization science and the 'CEO speak' of consultants, have had to deal with the problem of sharpening referential precision far more explicitly. And they have: we find (Tables 1A-1C) a host of complexity measures that *speak to different purposes* and call out different kinds of *difficulties* vis a vis a phenomenon: the difficulty of describing it (table 1A), the difficulty of synthesizing it (Table 1B) and the difficulty 9of synthesizing it from a description of known complexity (Table 1C).

The challenge for anyone who wishes to use the words 'complex' and 'complexity' in a way that disambiguates their referents (and it may be that not all, and not even many users of these words share this intent) is to either make clear the specific choice of definition of a complexity measure, or to create a measure of complexity that spans the range of options in a way that makes 'complexity' transparent in its usage. The path to the first option is self evident. We will pursue the second option here.

Table 1 A. ‘Complexity’ measures that Speak to How Difficult a Phenomenon is to Describe.

Measure	Explanation
‘Log-p Information’	Measures (in bits, if base-2 logarithm) the amount of information embodied in a point measure of the likelihood of a random variable.
(Shannon) Entropy	Measures (in bits, if base-2 logarithm) the amount of uncertainty involved in the probability distribution of a random variable
Algorithmic Complexity	Measures (in bits) the length of the shortest program that will specify the object as output.
Minimum Description Length	Measures (in bits) the minimum length of the hypothesis that best encodes the data.
Fisher Information	Measures the amount of information that an observable random variable carries about an unknown parameter of a distribution modelling the random variable.
Renyi Entropy	Measures (in bits, if base-2 logarithm) the diversity, uncertainty, or randomness of a system.
Code Length, Huffman Code Length, Shannon-Fano Code Length, Error-correcting Code Length, Hamming Code Length	Measures of the number of coding digits assigned to a message, according to different coding algorithms
Chernoff Information	Measures of the bounds of the sums of the distributions of independent random variables.
Information Dimension	Measures of the normalized entropy of finely quantized versions of the random vectors.
Fractal Dimension	Measures of the detail (complexity) of a pattern, as a ratio of the change in detail to the change in scale.

Lempel-Ziv Complexity	Measures the least possible number of steps in which a sequence can be generated, via the number and length of the repeated sections.

Table 1B. Measures that Speak to 'How hard a Phenomenon is to Synthesize'.

Measure	**Explanation**
Computational Complexity	Measures (in time, or space) the amount of resources required to run an algorithm.
Time Computational Complexity	Measures (in number of operations) the time it takes to run an algorithm that generates the object.
Space Computational Complexity	Measures the amount of memory space needed to run an algorithm.
Information-Based Complexity	Measures The branch of computational complexity that deals with the intrinsic difficulty of the approximate solution of problems for which information is partial, contaminated, and priced
Logical Depth	Measures (in number of operations) the time it takes to run the shortest program that will specify the object as output. The evaluation of the complexity ***relies on the choice of a model of computation***, which consist in defining the basic operations that are done in a unit of time.
Thermodynamic Depth	Measures (in bits) the amount of uncertainty for macroscopic states of physical systems.
Cost	Measures (in $) the amount of money used to produce something or deliver a service.
Crypticity	Measures the difference between a process's hidden state information and its observed information.

Table 1C. Measures that Speak to How Hard A Phenomenon is to Synthesize Starting from a Description of a certain Complexity.

Measure	Explanation
Metric entropy	Measures how the uncertainty or information of the system evolves (grows) in time, or under iteration for a map.
Fractal dimension	Measures of the detail (complexity) of a pattern, as a ratio of the change in detail to the change in scale.
Excess entropy	Measures how much additional information (in bits) must be gained in order to reveal the actual uncertainty (entropy density, hμ)
Stochastic complexity	Measures the fewest number of binary digits with which the data can be encoded (relative to a class of models).
Sophistication	Measures (in bits) the structural (algorithmic) information of a string, the minimum complexity of the best model for the string.
Effective measure complexity	Measures the relative memory required to calculate the probability distribution of the next symbol of a sequence.
Topological epsilon-machine size	Measures the number of states of an epsilon machine (the unique minimal representation of stationary stochastic processes whose states are the equivalence classes of infinite histories with the same probability distribution over futures),
Conditional information	Measures the expected value of the mutual information of two random variables given the value of a third.
Schema length	Measures the total number of nodes in the schema (a subset of strings with similarities at certain string positions)
Hierarchical complexity	Simon's theory of hierarchy: a large proportion of the complex systems we observe in nature exhibit hierarchic structure. On theoretical grounds we could expect complex systems to be hierarchies in a world in which complexity had to evolve from simplicity. In their dynamics, hierarchies have a property, near-decomposability, that greatly simplifies their behavior.
Grammatical complexity	The level of a type of grammar, in terms of Chomsky's 3 criteria (the class of language it generates, the type of automaton that recognizes it, and the form its rules must have).

What We Shall Do. We will cogitate on fundamental results from mathematics, computer science, and physics to develop a complexity measure that can be informatively applied to organizational phenomena regardless of the models and metaphors used to represent them. We will show this measure is quantifiable, invariant under actor-observer transformations (avoiding the subjectivity/objectivity traps), and maximally transferrable across a broad range of both organizational phenomena and ways of representing them. We put forth an explicit model of organizational complexity that *generalizes across different representational frameworks*. We will build on work that has conceptualized organizations as 'repositories of problem solving knowledge' [Marengo, Dosi,Legrenzi and Pasquali, 2000; Moldoveanu and Bauer, 2004; Moldoveanu, 2009;2017;2019] and linked the algorithmic structure of problems to the organizational structure of firms to articulate a complexity measure that is maximally 'framework-invariant' and that helps one bridge explicitly between problem-solving activities (which can be seen as instantiating 'algorithms' running on a suitable substrate) and organizational processes and phenomena – including organizational failures and sub-optimal adaptations to complexity. We shall synthesize a measure of complexity that transcends the boundaries of any specific disciplinary approach to organizational phenomena while retaining the relationships between complexity measures and the representations of phenomena to which they refer. The new framework will provide not only a new set of linkages among 'old' entities, but also one for describing organizational dynamics as a set of intelligible and possibly intelligent adaptations to complexity. We will make clear what can be meant by both complexity-driven organizational failures and 'complexity challenges' to executive function - and 'innovating in complexity space'.

2. **The Representation-Dependence of 'Organizational Complexity'.** The class of entities whose complexity can be directly measured has the essential property of *representationality*: they are models, renderings, simulations, algorithms, lists, etc – that individually or jointly represent objects, events and phenomena. When we refer to 'the complexity of X' $(C(X))$ we (can only properly intend to) refer to 'the complexity of representation R of X – $(C(R(X)))$. To sharpen this point, consider the problem of measuring the 'complexity of a car'. Whatever route to the analysis of the object 'car' one takes - as a network of discrete elements such as masses, springs and dashpots, energy sources and sinks, or as set of closed systems of

energy flows, or as a set of design rules applied to assembly schemata, or as a set of mechanical and electrical sub-assemblies that can be combined according to specific compatibility conditions - one must commit to a *representation* of the car before one can talk about the 'complexity of the car.' To get a complexity measure from a representation, we need to tailor the complexity measure to the representation:

- For a set of sub-assemblies model of the car, one can define complexity as some multiplicative or additive combination of the number of sub-assemblies and the number of links between them.
- However, if the sub-assemblies have complex interfaces that take a long time to program, debug and integrate, then one might take that fact into account and factor the computational complexity of fitting two components together into one's complexity measure.
- If we want the 'complexity of the car' to include the difficulty of designing and manufacturing it, then we might also focus on the algorithmic complexity of the design and manufacturing processes – the number of elementary cognitive and physical operations these processes minimally entail.

The representation-dependence of the measurement of the complexity of objects and phenomena is troubling. It does not readily allow observers to reliably communicate with one another about the complexity of objects phenomena in a language that picks On the contrary, it engenders and fosters referential confusion: person A will say a car is complex - and mean 'complex' in a substructure-dependence sense while person B will say it is complex – and mean 'complex' in an algorithmic sense - and neither will, *sans* further elucidation, know what the other 'meant'. This state of affairs may be acceptable if the two definitions are strongly positively correlated across a wide range of systems. But they are not. The problem comes to light most forcefully when one theorizes about the 'complexity of an organizations' products' and wants to compare the complexity of a car with the complexity of a computer-aided design program. In spite of the fact that interesting – and several, in each case - complexity measures have been developed for 'cars' and 'design programs', these complexity measures are different among them, and do not allow researchers to bring together car-complexities with design-program-complexities under the umbrella of a theory or argument purporting to capture something fundamental about complexity

Despite the increasing interest in organizational complexity (e.g. Etilraj and Levinthal 2004, McKelvey 2004, Rivkin 2000, Levinthal and Warglien 1999, Brown and Eisenhardt 1998, Westhoff et al. 1996, Kauffman 1988), the definition of 'complexity' remains tied to particular frameworks which may equate it with 'non-obvious order', 'irreducible disorder' or various combinations thereof. Complexity has been defined by appeals to chaos theory (Thiétart and Forgues 1995) or the theory of complex adaptive systems, in particular to genetic algorithms (Holland 1975, Holland et al. 1986) and to Kauffman's NK(C) models and Boolean networks (Kauffman 1993, McKelvey, 1999, Rivkin, 2000). The use of the attribute 'complex' in these cases carries with it the descriptive apparatus of the language system that generated its specific usage:

- An organization is 'complex' in the sense of 'chaotic' if the dynamic behavior of some of the variables that describe ot instantiate the sensitive dependence on initial conditions characteristic of the dynamics of systems described by specific nonlinear ordinary or partial differential equations – *without, however*, having any good reason to believe that organizations can or should be described by such equations or others in their class.
- An organization is 'complex' in the sense of 'made up of a large number of tightly coupled interacting components' if there exists some representation (networks of influence or friendship, technical or informational dependencies among structural units, etc) that can be validly predicated of that organization, but the complexity measure – and the specific 'complexity' value it returns will depend on the observer's choice of both the specific modeling approach (organizations as *networks* of entities) and the specific choice of variables that are measured (networks of information flows, networks of influence, networks of value linked activity chains, etc.).
- An organization is 'complex' in the sense of 'random' or 'unpredictable' to its CEO if the set of predictive models and representations that she is using to predict the consequences of her and others' actions to a set of variables critical to the survival of the organization reliably and unexplainably fail to produce predictions of useful levels of accuracy.

The first step towards a solution to the problem of the representation-dependence of complexity is to accept that measuring the complexity of an object or phenomenon requires we first represent it using a specific model, schema or 'language', and only then calculate its complexity by computing the complexity of that specific representation.

Of course, we are after a way of measuring complexity that is independent of the precise representation we choose. The basic method by which we compute the complexity of ("car, represented as network of discrete mechanical components") should be the same as that by which we compute the complexity of ("car, represented as procedural difficulty of design and manufacturing steps") and should also be the same as the basic method by which we compute the complexity of ("AcetylCoEnzyme A, represented as network of chemically interacting organic sub-assemblies and reaction sequences"). This does not mean the *value* of the complexity measures computed by this method in the three different cases will be the same; in fact, differences in the specific values of complexity turned up by applying the same method of computing complexity to different representations of the same phenomenon or object will be interesting to researchers and useful to practitioners.

The second step in our quest for a transcendental complexity measure is to recognize we are interested in *maximally efficient representations:* If we want to find the complexity of the representation, via ordered pairs, of a straight line segment given by $\{(1,1), (2,3), (3,5), (4,7),(5,9),(6,10),(7,13),(8,15),(9,17)\}$, we will first compress the data describing the segment to $(x,y{:}y=2x-1;\ x \subset N)$ (which is more compact than the representation based on enumerating the ordered pairs) and then measure the complexity of the (compressed) representation. The compressed representation does not immediately and self-evidently reproduce the original representation: It must first be fed into a computational device that can interpret it as a set of instructions which, when executed, produce the representation it had compressed. A true complexity measure will also take into account the difficulty of producing the uncompressed representation from the compressed representation – the number of operations this task will require of a machine or other entity capable of algorithmic operations.

Complexity as Informational Depth and Computational Load. We speak of the complexity of an organizational phenomenon as the complexity of a set of observations or models we choose to use to represent it. These observations amount to a digital object that can be represented in binary form and whose complexity we decompose into two parts:

The informational depth of a phenomenon – also known as 'algorithmic information' and hereafter abbreviated by 'I' – represents the amount of information contained in the representation of a phenomenon and is measured as the length – in number of bits – of the shortest program that can simulate the phenomenon (by reconstructing or synthesizing its representation) (Cover and Thomas 1991, Li and Vitányi 1997). Measuring informational depth is a two-step process. The first step (representation) maps the phenomenon (a set of observations) onto its description (a digital object), i.e. onto a sequence of symbols that defines (what is included in) the phenomenon under study. The second step, reduction (or, compression) maps the description into the shortest program that, when run on a computational device, reconstructs the digital object. Thus, the second step maps a digital object onto another; it maps the description onto a program that contains all information necessary for a computational device to generate the digital object. As many such programs are possible, it maps the description to the shortest program.

By definition, the shortest program that reconstructs a given digital object contains no redundancy: the shortest program that can simulate a certain digital object is a sequence of symbols that cannotbe compressed any further. It is 'algorithmically random': no substring contains any information about the rest of the string [Cover and Thomas, 1991]. Thus, the length of the shortest program capable of reconstructing a particular digital object establishes a lower bound for the algorithmic information content of this object. It has been proven impossible (Chaitin 1974, 1982, Li and Vitányi 1997), to rule out with certainty the possibility of progress in the sense of a pattern being discovered in a string heretofore considered random (i.e. incompressible). The string '0110101000001 001111001100110011111100111011' seems random to a lay person; some mathematicians, however, recognize this string as the first 42 digits of the binary representation of $\sqrt{2}-1$ and, therefore can replace a long string with a short program that calculates $\sqrt{2}-1$.

Measuring the informational depth of a phenomenon as the length of a random bit string seems counter-intuitive but there is nothing paradoxical about this approach: A scientific theory is comprised of theorems, axioms, and inference rules. Its theorems are reducible to its axioms – the basic statements from which all other statements can be derived. Its axioms and inference rules, however, are *irreducible*: they cannot themselves be derived from other statements that comprise the theory in question. They represent that which we must take for granted in a theory. Analogously, reducing a phenomenon to a random bit string that, when run on a computer, can simulate the phenomenon, identifies the amount of information that cannot be derived and therefore must be listed (as a sequence of independent bits that make no reference to each other).

The computational load of a phenomenon – also known as 'time complexity' and abbreviated by *'K'* – the total number of operations or the total time – required to execute the program that produces the representation of that phenomenon as its output (see, for instance, Garey and Johnson, 1979; Cormen, Leiserson and Rivest, 1993; Papadimitriou, 1994 for discussions of computational complexity or time complexity, both of which are equivalent to 'computational load' in our sense). Partitioning the information required for simulating a class of phenomena into 'input' (the parameters of a specific element) and 'algorithm' (the procedure applying to the entire class) makes it possible to infer the computational load of a digital object from knowledge of the computational load of other objects *in the same class*. In the case of matrix multiplication, computational load grows quadratically with the size (number of cells) of the matrix. Efficient algorithms for sorting and searching decision trees (which can simulate various categorization and prioritization tasks) require run times that grow logarithmically as a function of input size (Cormen, Leiserson and Rivest 1993). The paradigmatic example of problems for which no tractable (*P*-hard) solution algorithm isknown to exist is the Traveling Salesman Problem (TSP) (Garey and Johnson 1979) and is representative of many managerial logistics problems [Moldoveanu and Bauer, 2004]. Examples of other organizational or managerial problems that have been shown to be *NP*-hard include: strategic analysis and competitive decision making (Moldoveanu and Bauer 2004), diagnostic reasoning (Miller et al. 1982), software design (Abelson and Sussman 1984), engineering systems design (Chapman, Rozenblit and Bahill 2001), and the discovery of equilibria in competitive games (for n players choosing their strategy depending on the other players' choice) (Gilboa 1989).

Transcendental (K,I-Space) Representation of Organizational Complexity. We are now in a position to put together the two dimensions of complexity into a single, non-separable algorithmic procedure we shall call COMP, and we define based on a refinement of Bennet's [1988] insight on combining informational and computational dimensions of complexity. COMP operates as follows (Figures 1,2):

a. for a given digital object (a finite representation of a phenomenon or object) *O*, a chosen computational device (*Turing machine) TM*, and a particular maximum number of computations K_{max}, *COMP* outputs the informational depth $I(O, TM, K_{max})$ of the program *P(O)* that can generate O on TM using up a maximum of K_{max} operations TM if such a program exists, or returns an error message if such a program does not exist;

or,

b. for a given digital object (an in-principle computable representation of a phenomenon) O, a chosen computational device (*Turing machine*) *TM*, and a particular maximum number of bits of program memory – or, a maximum informational depth I_{max}, COMP outputs the computational load $K(O, TM, I_{max})$ of the program $P(O)$ of maximum size I_{max} that can generate O on TM iff such a program exists, or returns an error message if such a program does not exist.

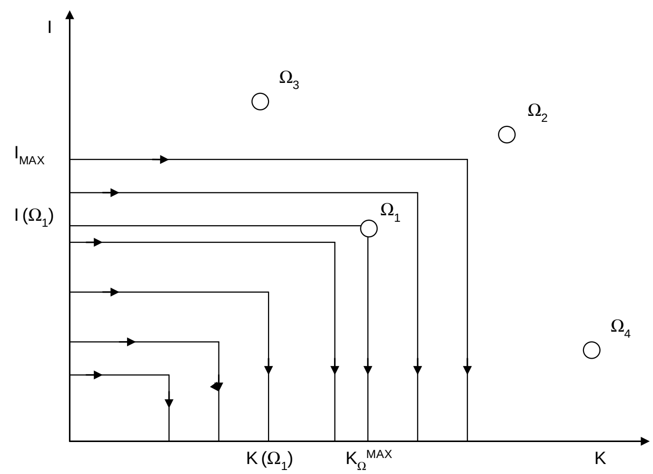

Figure 1: I-Driven Search for the Complexity of a Representation. Ω_2, Ω_3, Ω_4 are "invisible" objects to COMP,* which is constrained in its search by I_{MAX}, K_{MAX}

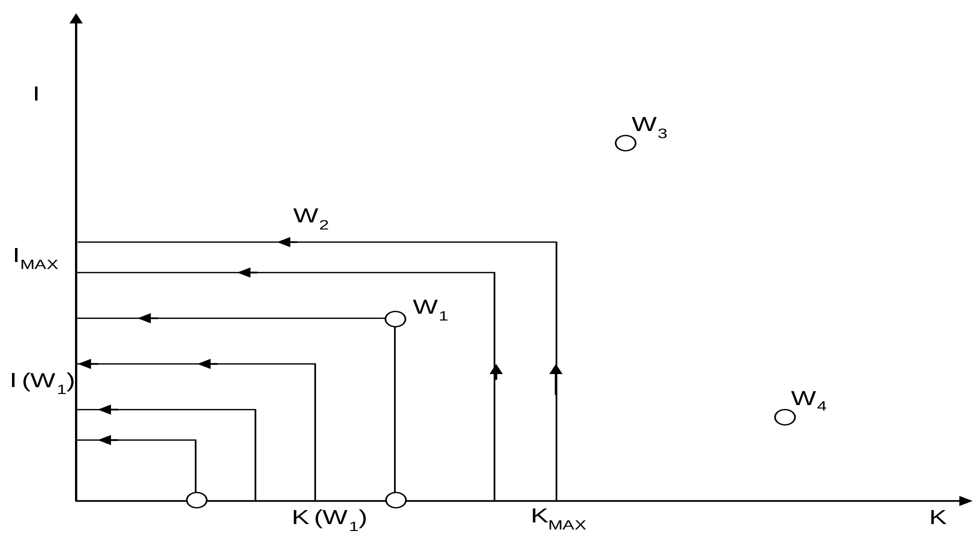

Figure 2. K-Driven Search for the Complexity of a representation.

`Both instantiations of COMP require that it find the program P that satisfies the requisite I-level or K-level constraint if P exists. This makes it possible to interpret an error message as proof for the non-existence of P conditional on having decided on the computational machine *TM* that will perform the search.

In its first instantiation, *COMP* searches for *P* by going through all of the possible bit strings that can function as input to *TM* and returning an *ERROR* message if O cannot be computed in fewer than K_{max} operations, or the pair, *(I(O), K(O))* otherwise, corresponding to the informational depth and computational load of synthesizing O on TM.

There is no upper bound on the search space: a (potentially) infinite number of bit strings may have to be searched, and therefore COMP in the first instantiation is not itself computable. COMP can search for P in the second instantiation by going through all of the bit strings of length less than or equal to I_{max} (there are at most 2^{Imax} of them) and measuring the computational complexity K of each program. However, because there is no *a priori* constraint on K, there is also no guarantee that COMP can find P in a finite number of operations: we do not know the maximum amount of time we have to wait for finding P, and therefore we cannot infer the non-existence of P from not finding an I_{max}-long program P that does not converge to O after k iterations, for any finite k. Therefore COMP in the second instantiation is not computable either, and there is no guarantee that COMP will return P satisfying constraints on either I alone or K alone, given that P exists.

Given this challenge, we propose a modified definition of *COMP* (and call it *COMP**), which is computable at the cost of constraining the search space in which we look for the complexity of an object *O* and therefore eliminating the logical link between the existence of P and its discovery by *COMP**. In the new definition, for a given digital object (a binary representation of a phenomenon) *O*, a chosen computational device (Turing machine) *TM*, and a particular maximum number of bits of program memory – or, a maximum informational depth I_{max} and a maximum number of operations K_{max}, *COMP** outputs the computational load and informational depth *(K*(O, TM), I*(O, TM))* of the program *P*(O)* of maximum size I_{max} that can generate O on *TM* in K_{max} operations or fewer iff *P** exists, or returns an ERROR message iff P* does not exist. *COMP** searches for the complexity of O in a window that is determined by the dimensions (K_{max}, I_{max}). This definition of the 'complexity search function' makes the (resource-bounded) complexity of a phenomenon (that can be represented by a binary object O) a computable quantity, and makes precise the assumptions that go into assigning a complexity measure to a phenomenon and the limitations of any possible complexity measure:

a. the complexity measure $(I^*(O), K^*(O))$ is relative to a particular Turing Machine (*TM*). This is a known result in complexity theory [Li and Vitanyi, 1993]. In our context, this assumption is important because it highlights the importance of being precise about the underlying modeling language in which a phenomenon is represented (i.e. the language in which P is written);

b. the complexity measure $(I^*(O), K^*(O))$ is relative to a particular window, (K_{max}, I_{max}) that limits the search space. Some program P^{**} may have lower informational depth than $I^*(O)$; however, if they entail (on *TM*) a computational load greater than K_{max}, then $COMP^*$ will not find it, as $COMP^*$ only looks in the window (K_{max}, I_{max}). This makes our estimates for the complexity of an object (and its associated phenomenon) dependent on the current state of the art of computational speed (related to K_{max}) and memory density (related to I_{max}).

4. *Organizational Dynamics in Complexity Space: 'Coping', 'Innovating' and 'Failing'.*

We model organizational dynamics in the 2D complexity space introduced above by representing organizational phenomena in terms of their informational and computational dynamics. The aim is to synthesize a space for representing organizational structure and dynamics that functions at a level of abstraction similar to that of a *phase space* in classical and quantum mechanics [Thorne et al, 2018]. The value of such a space in the modeling of physical systems, for instance, is that it allows the modeler to predict and control the dynamics of large numbers of systems that are equivalent to one another on the basis of the topology of their phase spaces [Figure 3]. It would surely be of great interest to arrive at a similar classification of interpersonal and organizational phenomena on the basis of their canonical dynamics in complexity space.

We model organizational behaviors as mechanisms that operate *on* complexity: as mechanisms for reducing, increasing maintaining one or both values of our complexity measure for a set of phenomena conditional upon a set of agreed upon representations thereof. [1] The organizational costs associated with the informational depth and computational load of its phenomena are not immediately obvious. They need to be specified. We do so by asking: what is the problem that the organization must solve to successfully carry out a high *I* - high *K* task?

a. *The costs of informational depth*. High-I tasks confront the organization with the problem of *beholding*: a minimal representation of an activity must be beheld (collectively or individually) in order for the activity to represent purposefully pursued organizational behavior and the informational depth of a task is the minimum amount of information that the organization must behold, as it is *the* minimal representation of the behavior in question. Discovering or creating the required information, embodying it (and thus turning it into action) and encoding- decoding it (for the purpose of storage/recall and communication-coordination) are all constitutive components of beholding and thus can be understood as the core sources of *I-related organizational costs*.

b. *The costs of computational load*. High-K tasks confront the organization with the problem of *executing*: *K* measures primitive operations (or, minimal behavioral steps) that the organization must go through in order to carry out a task. Given a finite total number of operations that comprise a particular task, one can improve an organization's ability to carry out the task satisfactorily by increasing the frequency with which basic operations ban be performed ('speed'), the number of people and machines that are carrying out the task in parallel ('resources') and the time available for the task to be completed ('duration'). Accordingly, speed, resources and time are the critical components of *K*-related organizational costs.

To highlight the use of this complexity-space approach and its relative representational and modeling power – let us focus on a familiar example in the literature on organizational complexity: Herbert Simon's picture of two watch

---

[1] Executives can rationally choose to increase the complexity of an overall predicament fpor themselves and others: A laggard in a solidifying industry, for example, can act to throw an emerging industry standard in disarray, increasing the complexity of the predicament for every other firm in the industry (including itself); and its actions can be rationalized, as it can do no worse than finish last, which is the most likely outcome under current conditions in any case. A top management team member can choose to undermine a particular strategic agenda by challenging closely held assumptions on which the agenda is based without advancing alternative assumptions on which a new agenda can be built, thus creating high levels of complexity along both the dimensions of informational depth (as 'no plan' is always equivalent to the entire gamut of raw data and imperatives that managers regard as true or useful) and computational load (in the form of logistical tasks of setting up new planning meetings, governed by new agendas and decision making processes, etc.) imposes a high immediate cost of execution.

makers (Hora and Tempus) each tasked with making a 'complex' (i.e. many parts, many links between them in Simon's (1962) language) watch of about 1000 parts. Simon used this picture to argue that complex, purposefully guided systems 'evolve' towards hierarchical, modular architectures. Tempus assembles watches using a 'serial' process, assembling one watch at a time by putting together parts 1 through 1000 according to a specified plan. Hora, on the other hand, creates intermediate assemblies of about 10 parts each, which she then puts together into incrementally larger assemblies, until a full watch is completed. Simon makes a special assumption about the influence of the environment: if either watch maker is interrupted while putting together an assembly, that assembly is lost. He shows that even for very small probabilities of exogenous 'interruption', Hora will fare considerably better than will Tempus (assuming each has the goal of putting together as many correctly assembled watches as possible in a given period of time).

The (*K*,*I*) formalism allows us to ask - and answer questions that Simon himself did not because he could not:

- What is the optimal level and degree of modularization of a complex task or product?
- What are the costs and benefits to Tempus to changing his/her approach to making watches to resemble those of Hora, given that he already has started down the path of serial assembly?
- If Tempus grows to a (multi-employee) Tempus, Inc. and Hora similarly turns into Hora, Inc. employing many different watch makers, what additional complexity-driven effects appear?
- What are the costs and benefits of modularization in this case?
- Besides the costs imposed by high numbers of components and high numbers of links among them, what are the ((*I*, *K*)-space) complexity costs of Hora Inc. and Tempus, Inc.?
- Other than modularization, what are the complexity coping mechanisms that Tempus, Inc. and Hora, Inc. can deploy to mitigate these complexity costs?
- What are the complexity costs of the actions involved in minimizing the complexity costs of a particular task and how should they be incorporated in the representation of a reflexive organizational complexity coping mechanism?

We partition organizational complexity coping mechanisms into three distinct classes:

- First, organizations can reduce complexity by looking for *redundancies, short-cuts and optimal trade-offs* between I and K. These measures effect direct cuts in the complexity measures of organizational tasks and behaviors, with attending reductions in complexity costs.
- Second, organizations can reduce complexity by *changing criteria* of optimality and convergence of a particular sequence of tasks; they can choose to settle for *approximate* solutions to organizational problems and *approximate responses* to the market requirements for a product, or to settle for *probabilistic* and *statistical* rather than *deterministic* performance to a specified set of product, process or procedural standards.
- Finally, organizations can choose to re-design and re-partition the *internal* individual, group or organization-level resources carrying out a particular task, by changing the boundaries of the organization, the boundaries of the group or (and) the (K,I)-profile of the skill and capability set of the individuals making up the group or team. Let us consider these mechanisms *seriatim*, with examples furnished by Hora, Tempus and their scaled-up counterparts, Hora, Inc. and Tempus, Inc.

Here they are, *seriatim*:

Seeking Short Cuts: Changes in Process, Representation and Trade-offs Between I and K. Perhaps the most commonly occurring family of complexity coping mechanisms comprises attempts to exploit local redundancies – either functional ones, providing reductions in computational load (K), or representational ones, providing reductions in informational depth (I) – and attempts to effect optimal trade-offs between I and K.

K-Wise Search for Most Efficient Algorithms: Exploiting Functional Redundancy. Functional redundancy can be conceptualized in (*K,I*) –space as a set of steps that are an integral part of a task, but are – under some representation of the task – redundant. Suppose that Tempus drills 2 micro-machined parts on a high-precision machine for every watch assembly that she puts together, then walks away from the machine to go on to preparing the next part. During this time, he must find a place (dust and static-free) to store the micro-machined parts. The storage operation, along with the setup and shut-down time for the micro-drill become part of the K-count of his watch making process. More functionally minded, Hora decides to drill the micro-machined parts in batches of, say, 200, corresponding to one week's supply of watches, and thus saves 100 times the cost of machine set-up and shut-down steps and part storage steps that Tempus must incur. Exploiting K-wise redundancy has to do with seeking a *partitioning of the production function that entails the fewest number of steps*. This is often achieved through the basic operations of pooling parts and pooling inventories of finished product, whose costs (the expected value of the loss associated with part and finished product inventories) must be traded off against the cost savings due to *K*-wise minimization.

I-Wise Search for Most Efficient Algorithms: Exploiting Representational Redundancy. Finding 'the right' representation of the production task is equally important for the minimization of the informational depth associated with that task. *I* is only *provisionally* the most compact representation of a task. (It is minimal until another, more efficient representation is found). While there exist some hard limits on representational compactness (for example, no classical 2-bit representation of the decimal number 9 is achievable, regardless of advances in representational 'technology'), these limits are not easy to come by (as there exists no algorithm that can discover them for any representation) and are not obvious upon inspection of a particular task enunciation (for example, is the *General Theory of Relativity* minimal in the *I*-sense?) Looking for 'short-cuts' in *I* may mean either looking for redundant symbols (for example: 'e', 'u', 'an', 'y', 'l' can be understood as redundant to 'rdndt smbls' in the unpacked representation redundant symbols – where the vowels and some consonants serve as 'error-correcting devices' increasing intelligibility in spoken and written language), or looking for languages that provide the minimal representations for a particular task (for example, the mathematical language of integral calculus, $\iiint_V F(x,y,z)dxdydz$ is informationally more efficient than the natural language representation ('divide up the closed volume *V* into small volume elements with sides *dx,dy,dz*, evaluate the function *F(x,y,z)* at each point inside the volume *V*, multiply the function at each point by the volume *dxdydz* in the neighborhood of the point at which *F* is evaluated, take the limit as the volume element in question tends to zero of the sum of all of the volume elements in *V* multiplied by the volume elements in question') of the exact same operation (to someone or something equipped with the semantic conceptions corresponding to the words and symbols in question).

Suppose both Hora and Tempus must face the problem of storing many different watch designs - which they have to produce quickly, upon demand. Tempus – in linear fashion – takes the approach of storing – in a file drawer or a digital computer – all of the watch designs (say, 1000) that he feels he needs. Hora stores only a few watch designs and then stores only the differences between these watch designs and the remaining designs. Assuming that watch designs are contain some levels of redundancy, the informational storage requirements of Hora will be significantly less than those of Tempus.

To show how the representation-dependence of efficiency (or, complexity) figures into the watch design and manufacturing process, suppose that Tempus uses encoded bitmap images of the various watch designs in a large computer memory. The total number of bytes that Tempus needs to store is equal to the number of bytes per bitmap (*1000 x 1000pixels x (16bits/pixel)/(8bits/byte)))~2MB* of a design multiplied by the number of different watch designs (1000), giving *2000MB - 2GB*. Hora passes her watch designs through a program that generates a Bezier spline (a special, highly flexible function used to efficiently represent 3-dimensional surfaces and textures in memory-intensive applications such as video games) representation thereof, storing only the 'Bezier coefficients' associated with each design (up to 1000, times 16 bits each) plus a table of the used Bezier splines (up to 1000, each of which can be compressed to a 1000-bit-number algorithmic representation), and, accordingly, needs only approximately 128KB to store a single watch design and therefore just over 1 MB to store the entire watch design collection, as compared to the 2GB needed by Tempus for the same task.

We should thus expect organizations to develop internal *languages* and *codes* (Arrow 1974) that serve the purpose of decreasing representational complexity (or, informational depth in our language). These representation *technologies* need not be *technically sophisticated* – as were our examples; indeed, the density of acronyms and terms of art in law and medicine and the 'euphemistic' languages used by strategy consultants and professors of strategy can be seen as representational languages (in the case of 'business talk') and codes mapping observational language into a more compact representation (in the base of law and medicine) seems to us to indicate the presence of an overarching *compression heuristic* enjoining the generation of increasingly compact representational representations (in the minimal-I sense) over time.

Optimizing Trade-offs between I and K. To see how informational depth can be traded off for computational load, consider two processes for finding out the outcome of a coin toss on a soft surface. The first (high *I*) relies on a precise measurement of the initial position, upward acceleration, height from the ground and angular momentum of the coin, followed by the computation of its precise trajectory using a model based on Newton's laws. The second (high *K*) relies on performing many blind coin tosses and measuring the relevant heads/tails frequencies, then making (the usual, somewhat suspect) assumptions equating (finite) frequencies with (approximate) probabilities to derive 'the probability of heads' there-from. In the second case, I (which in the first example is embedded in the accurate representation of the initial conditions of the coin and the model for tracing the trajectory of the coin through the air) is traded off for a large number of locally 'dumb' operations (coin tosses) which, in the aggregate, can be said to provide the modeler some intelligence regarding the outcome of a future toss. Depending on the marginal costs of *I* and *K* encountered by organizations, we would expect to be able to refer to parts of the activity sets as attempts to effect optimal trade-offs between the informational depth and computational load of their tasks.

Suppose Hora could achieve another tenfold efficiency in the representation of the watch design library (over that he already has achieved relative to Tempus) by using a set of basis functions that are 'self-similar' – they are identical to themselves at ever smaller spatial scales, and therefore have a 'fractal' structure (Barnsley 1986). Once such a fractal kernel is found for representing an image, very large compression ratios are achievable. However, since each image has 'its own' fractal kernel that must be constructed 'from scratch', the computational load of the compression process far exceeds that of any other compression process. Should Hora dispose of large computational devices and only limited memory, however, it would be reasonable for her to attempt to trade off greater computational load for lesser informational depth.

Trade-offs between I and K are to be contemplated by any organization facing the choice between compressing its explicit knowledge base to a minimal set of rules and imperatives that are 'common knowledge' (in the technical sense that everyone knows them, knows everyone knows them, and so forth [Lewis, 1969]) among its members who can 'figure out' (by performing high-K tasks) all of their local implications and count on others to do the same and creating pools of impressions, factoids, 'stories', myths and metaphors that are not logically connected and are less likely to collectively become 'common knowledge' but provide richer local cognitive and intuitive maps to those that possess them. Even though a classification of organizational knowledge on the basis of its 'axiomatizability' and actual degree of axiomatization is beyond the scope of this paper, it is nevertheless a task that the (*K*,*I*) space analysis of productions tasks unveils as significant (and achievable).

Changing the Solution or Optimality Criterion: Approximation and Randomization.

Approximation mechanisms loosen the selection criterion for what an acceptable solution is: they allow Hora and Tempus to build "90%" of a watch, 100% of the time". Randomization mechanisms loosen the constraints on the statistics of an acceptable solution. They allow Hora and Tempus to build a "100% watch, 90% of the time". Both mechanisms are used by computer programmers to cut through to a solution of an unmanageably complex design engineering problem (Cormen, Leiserson and Rivest 1990). Here is how they work in organizational settings as complexity coping mechanisms:

Loosening Optimality Constraints: *The Logic of Approximation*. Approximate solutions are solutions that get designers of complex objects to within an acceptable distance of a finished final product. A linear approximation to a nonlinear function is an approximation of the nonlinear function that has the complexity of a linear function. It is usually good over a specified range of the input variables. For example, the variable X is a linear approximation to the variable X^2 for values of X around 1 (where $X=X^2$), and, more generally, LX is a good linear approximation to the function LX^2 for values of X equal to l/L. As long as X is constrained to take on values that cluster around the specified limiting input values, the value of the output F(X) can be constrained to take on values that do not deviate by more than a specified amount from the exact value of the nonlinear function.

Consider the *design* problem that Hora and Tempus face, i.e. the problem of producing a watch that maximizes an objective function (performance, for instance, measured in deviations of the time the watch shows from the exact time measured by an atomic Cessium clock that sets the standard for accuracy) given a large set of input constraints (factor costs, tolerance levels on parts that bear a particular acceptable cost). This problem can be represented as the 'knapsack problem' (filling a knapsack as closely as possible to its capacity with utensils that have pre-determined expected utilities), which is known to be *NP*-hard – explicitly to computer scientists and implicitly to engineering designers and developers. Hora has modularized her design process and breaks down the watch into 10 distinct sub-assemblies of 100 parts each - individually optimized. Then, assuming that the complexity of the design problem is exponential in the number of parts that must be taken into account in the design process, Hora will register a reduction in the computational complexity of her design task of $2^{1000}/(10 \times 2^{100})$, *or* 0.1×2^{10}, or, roughly a 100-fold decrease in K as a result of the breakdown of the 'problem space' into 10 individually tackled 'sub-problems'.

The resulting solution may not be optimal relative to the solution of the full optimization problem, which contains *1000* different variables. Some 'global maxima' may be missed as a result. If the overall objective function is reasonably smooth, however, and therefore if the local maxima picked up by the 10 parallel 100-variable optimization processes do not deviate from the global maximum by more than an amount that is acceptable to the designer, then the trade-off between decreased complexity and decreased accuracy may well be worth making for Hora. Simon's modularization logic with respect to *making* has a similar mechanism behind it: through modularization, Hora achieves a decrease in the expected value of the number of operations that she will have to perform to manufacture one watch in the presence of environmental interruptions. However, our example illustrates the advantages of modularization even in cases and situations where there are no environmental effects that drive the logic of modularization in a complex system.

Loosening Deterministic Convergence Constraints: *The Logic of Randomization*. Another route to complexity coping is that of loosening constraints on the statistics with which the 'exact' answer or the globally optimal solution can be reached. Randomization techniques rely on disciplined guessing strategies that trade off statistical certainty for decreases in complexity. For example, a polynomial-time approximation to the answer to the Traveling Salesman Problem can be achieved at various levels of accuracy and reliability (eg: the probability that the solution achieved by intelligent guessing is indeed the optimal one) by using local searches around random guesses at the minimum path connecting all of the points in the network of cities or places that must be connected by a minimum-length path.

Hora has now become Hora, Inc. - and Tempus has become Tempus, Inc. They now face the problem of mass-manufacturing their products on large assembly lines. In setting up these lines, they must solve TSP-like problems in setting up their assembly equipment (to minimize the distance that a watch or sub-assemblies of the watch must spend traveling around the shop floor before final assembly). Suppose Tempus, Inc. attempts to solve the problem by systematic searches among all of the possible allowable trajectories that a part must take around the shop floor: it then must solve the full TSP in order to get to the optimal solution. Hora, Inc., enlightened by the logic of randomization, searches along randomly chosen (or, pseudo-randomly generated) initial paths around the shop floor. Depending on available resources, its managers control either the *depth of the local searches* or the *number of initial starting paths*. Hora has thus created a non-deterministic search environment that has a good chance of converging to the optimal solution, without incurring the full cost of solving the (intractable) problem Tempus must solve. It has traded off certainty of optimality for lower computational load of the search process. Given that many such organizational search processes are (a) intractable [Rivkin, 2000; Moldoveanu and Bauer, 2004] and (b) subject to the kind of environmental interruptions that Simon introduced to motivate his original work on complexity, the probability of completing a global search process 'before interruption' will often be less than the product of the probability of completing the local search processes before interruption and the probability that these search processes will lead to the optimal solution, which will become, for organizational designers, the 'decision variable' that governs switching between full/deterministic and local/probabilistic searches.

Taking Optimization to Higher Levels of Awareness: Complexity Bounds on Complexity Coping. Consider the following hierarchy of problems that Hora or Tempus can – and perforce do - face:

Problem P: optimizing the watch-making process given a watch design;

Problem PP: optimizing the watch design process given a set of desiderata about and constraints on the product;

Problem PPP: optimizing the joint watch design and manufacturing processes given a set of desiderata and design constraints;

Problem PPPP: optimizing the process of optimizing the watch design and manufacturing process given a set of desiderata and design constraints;

Problem PPPPP: optimizing the process of optimizing the process of optimizing the watch design and manufacturing process given a set of desiderata and design constraints;

Problem PPPPP… . optimizing the process of ……

What is a rational or reasonable end-point for this series of problems? Is any stopping point as good as any other? If not, is there a way of deriving a 'ranking' of various alternative stopping points?

Reflexive entities like individuals and organizations can and often do turn optimization problems (as well as 'justification problems' and 'argumentation problems') into problems of *infinite regress*. Once catalogued as such, they become 'uninteresting' and potentially even 'uninteresting' as objects of serious study. Many problems of infinite regress, however, can be rationally and even optimally truncated, provided that we can represent them in an appropriate space. The *(K,I)* space that we have introduced provides such a medium: Given that a design (or, optimization) problem has a particular level of difficulty (represented by a point in (K,I) space), it is possible to predict when a reflexive, optimization-minded manager will choose to cut through the infinite hierarchy of problems outlined above through to a behavioral solution and 'just do it'. It is that point at which the difference of the expected benefits of carrying out this optimization and the ((*K,I*)-parametrized) difficulty (costs) of the higher level optimization problem will precisely cancel out the difference of the expected benefits 'just acting' on the basis of the current predicament and the (*K,I*)-related costs of that predicament.

Thus: Hora will undertake to optimize its design and manufacturing process if the expected benefits of doing so less the complexity costs of doing so will outweigh the expected benefits of going with its current plans less the complexity costs of going with its current plans. To estimate the complexity costs of an unknown or not-fully-articulated optimization problem, Hora can develop complexity-wise heuristics, or, heuristics about the complexity classes of certain problems. It may develop (perhaps not fully articulated) intuitions about the complexity of knapsack-like and *TSP*-like problems, or about the memory and information access requirements of certain design representations and languages for representation. These heuristics will become embedded in its 'design philosophy', which can now be precisely articulated as the regions in the space spanned by *I* and *K* that the organization will avoid through its planning processes. Thus, a global, infinite-regress-free optimization can be contemplated in the (*K,I)* design framework, provided, of course, that we forego the requirement for certainty about the optimality of the resulting solution.

Complexity-Driven Structuration: Changing the Structures Organizations Use to Carry Out Tasks.

Not all complexity coping is rationally planned adaptation to locally changing tasks and goals. Indeed, the literature on organizational structuration teaches that many organizational structures are 'sticky' and stable over long periods of time, as are organizational routines and heuristics. The purpose of this section is to provide a representation of organizational structuration approaches using the (*K,I*) framework, and to show how organizational structures, routines, heuristics and patterns of reasoning can be understood in a single, unified framework.

A computer programmer facing a difficult programming task is most often constrained by the hardware on which she operates. She may work in a multi-processor environment or a single processor environment, each of which will influence level of parallelization of the code that may be achieved. She may be bound to work in one or another of the structured computer languages (like JAVA, J) available and compilable on the hardware she uses. She may already have libraries of subroutines (such as those for computing the logarithm or sine or cosine of a number) that provide gains in time-to-development over the development of new routines 'from scratch', even if these routines may end up being faster or more accurate than the inherited routines. She may have certain mental habits and 'tricks' that she uses to program, of which she may not be conscious: these are complexity-coping mechanisms that have become part of her own programming 'style'. This example already canvasses most elements of a (K,I)-based model of the spatio-temporal structuration of organizations, as follows:

Spatial Structuration: Hard-wiring group and organizational boundaries. Organizations are 'always-already' structured: into groups, teams, functional units and divisions, centres of excellence and cost minimization centres. These units jointly provide the 'hardware fabric' on which organizational tasks can be seen as 'running'. In (*K,I*) terms, they represent the 'sunk costs' of previous structuration attempts, successes and failures. They give the status quo of the organizational processing environment, which imposes boundaries and constraints on the current complexity-coping efforts of the organization.

Suppose Hora, Inc. has inherited (from Hora's old one-man shop) the 10-assembly partitioning that was the basis of Simon's analysis [Simon, 1962]. Its current organizational chart is structured into groups specializing in the 10 assemblies, simply on the basis of tradition and precedent generating path-dependent dynamics. Individuals with specialized skills are hired into the groups corresponding to the assemblies in question. The groups adopt pay-for-performance schemas and implicit reward and recognition structures optimized for the tasks they individually carry out. The complexity coping *Spielraum* of Hora, Inc. – its space of playful possibility and exploration - is constrained by its 10-group contractual (implicit and explicit) structure in the same way in which a programmer's task is constrained by a multi-processing environment: it must take into account the (*K,I*) costs –along with all of the other, factor-price-related costs – of changing its structure around when entertaining new architectures, predicated on different structural partitionings of a watch.This problem will not beset Tempus, Inc. - provided it has not hard-wired modularity norms into its organizational structure - which may therefore be favored when it comes time to radically change design architectures with changes in demand or technology.

Temporal Structuration: *Hard-wiring behavioral routines and sub-routines*. Organizations are 'always-already' temporally structured into behavioral routines and sub-routines (Nelson and Winter 1982) that are also both temporally sticky and locally efficient. Hora, Inc. may have developed routines corresponding to local, individual-assembly-oriented manufacturing processes (such as quality control routines and rework routines for non-functioning assemblies), whereas Tempus, Inc.

may have developed routines aimed at fast processing of a single watch down an assembly line. These routines provide a functional partitioning of the organizational problem space that constrains further complexity coping, by lowering the costs of routine-compatible or routine-embedded activities relative to those of out-of-routine activities. Hora, Inc., then, is likely to systematically forego global design and manufacturing optimization exercises (which cut across its already developed routine sets) whereas Tempus Inc. is likely to systematically forego routine sets concentrated on assembly-wise optimization (which would cut across its global optimization oriented routine sets).

Socio- Cognitive and Linguistic Structuration: Hard-wiring optimization processes and representation languages. As Whitehead observed, human progress often consists of the rendering-unconscious of processes and activities that are consciously pursued (rather than the making-explicit of implicit knowledge, as claimed by [Nonaka and Takeuchi 1995] - a view that is perfectly consistent with the idea that unconscious mental processes are physiologically faster and metabolically cheaper than are conscious ones).

In organizations, what is habituated in this way are the cognitive routines and sub-routines - often referred to as 'heuristics' (Cyert and March 1963) and the languages and representations – communication codes, design categories and ontologies. Each of Hora, Inc.'s 10 structural groups, for instance, can develop quasi-private languages for referring to individual components and sub-assemblies of components that are optimally (albeit locally) suited to their tasks. They can develop ways of talking (acronyms, special languages) that are optimally 'designed' for the modular task environment of Hora, Inc. They can develop ways of interpersonal inter-relating (long group-wide planning and design meetings, etc.) that are propitious for both the nature of the task and the structure of the group. These ways-of-being (of thinking, talking, inter-relating) are likely to be different from those engendered by Tempus, Inc., where, for instance, the size of the group and the global nature of the design process make it impractical to hold practical group-wide meetings, which means that planning is de facto done through a number of ad hoc, small group meetings, which in turn are likely to generate a quickly shifting fabric of ways-of-talking, design ontologies and cognitive habits than those to be observed in Hora, Inc.All three forms of structuration (spatial through group composition and contract, temporal through routinization, social and linguistic through the development of interaction patterns and cognitive through habituation of conscious mental tasks) can in fact be understood as adaptive, complexity-driven moves in I,K-space:

a. Spatial partitioning of the organization is a way of hard-wiring certain levels of modularization of planning, design and manufacturing into the organizational structure itself (and thus foregoing the costly optimization step of figuring out 'the right' level of modularization for each new task or sequence of tasks). The high-level problem on deciding on the right level of modularity is thus 'cut through', leading to decreases in both I and K. Such structuration can achieve the de facto parallelization of tasks, a common strategy for dealing with high-K problems in computer and algorithm design.

b. *Routinization* functions in organizations in ways similar to those in which routines are used by programmers: as fast, dedicated processing units that are optimally matched to certain tasks. They provide local speed at the expense of flexibility to global changes in constraints and desiderata, and can be understood as boundedly optimal adaptations to high-K task environments.

c. *Ontological and ontic commitments, heuristics, linguistic tricks ad shortcuts and special languages*, as well as other mental habits can be seen as adaptations to both high-I and high-K environments. Ontologies and special languages can be seen as more or less well-adapted *compression codes* for large bodies of information needed for the successful performance of a task. They are the languages in terms of which I is instantiated: their *own* informational depth can only be analyzed in terms of their representation in other languages and ontologies and therefore can only be critiqued as being more or less informationally compact *from the outside*, i.e. relative to other languages and ontologies. Linguistic tricks and dialects can be seen as *compressed versions* of longer, natural-language or perceptual-language messages, whose informational depth can be criticized relative to other forms of referring and encoding based on the same ontologies and representational forms.

d. *Cognitive habits and heuristics* can be seen as high-K adaptation devices, accomplishing the K-reduction function that behavioral routines carry out in organizations. They cut through the complexity of organizational optimization process to 'good enough' solutions or to solutions that are 'on average' optimal (with an acceptable frequency). This suggests that managerial 'biases and fallacies' can be exculpated in two ways from the usual charge of 'irrationality' - as *approximations* and as randomizations. In the first case, we would expect that they will perform in individual cases of furnishing predictions or other empirical judgments slightly (but tolerably) worse than their normative counterparts. In the second case, we would expect that they will perform *on average* as well as well as their normative counterparts normally do in furnishing predictions and other empirical judgments.

5. Complexity-Driven Organizational Failure in I-K Space.

Many accounts in recent literature implicate complexity in the processes that culminate in organizational breakdown or failure:

- 'normal accidents' occur in organizations that can be adequately represented by numerous, tightly and densely coupled structural elements that are functionally relevant [Perrow, 1976; Weick, 1995];
- environmental perturbations can and are statistically likely to cause densely, tightly coupled activity systems to break down [Simon, 1962];
- tightly, densely coupled activity systems exhibit complexity-catastrophe-related failures as the number of links and nodes between the activity systems making up the organizations increases;
- the computational complexity of making competent predictions about the evolution of a tightly coupled activity system admitting of a Boolean network model becomes prohibitive for even moderate levels of average link density between the elemental activities making up the organization [Rivkin, 2000] diminishing the ability of managers to 'plan' and thus adapt strategically (rather than reacting blindly) ;
- mismatches in 'fundamental variety' – or, degrees of freedom – between the organization and the part of the environment that is relevant to it limit the ability of the organization to new environmental conditions [see Scott, 1985, for review]; 'noise' and 'error' are amplified by computationally complex, serially coupled organizational production functions, decreasing survivability of value-linked activity chains in both the short run and the long run [Moldoveanu and Bauer, 2004; McKelvey, 1999].

Difficulties of planning and purposive action attributable to risk, uncertainty and ambiguity [March and Olsen, 1972] – the normal 'fog of business' – can also be seen as being causally linked to complexity – specifically to the barriers erected by the complexity of either carrying out experiments or calculations aimed at creating or refining probabilistic estimates that enable or encourage action and thus redress the paralysis that unknown-ness and unknowingness can cause. In spite of the insight afforded by these approaches into organizational breakdown and failure, they turn up several important shortcomings:

A. they are based on particular underlying models of organizations (as structurally coupled functional activity systems, or value-linked activity chains; as algorithmically linked production functions, as information processing devices with 'wired-in' representations for unknown-ness and unknowing-ness) and therefore are applicable *within* the representational space created by these models They do not – because they cannot - therefore capture the problems created by difficulties in applying the models to phenomena (either constructively or descriptively);

B. they therefore cannot easily be compared amongst themselves with regard to their predictions in specific cases. A particular level of complexity in one framework may entail a totally different level of complexity in another framework (consider value-linked activity systems described at different levels of resolution) and the same metric can have different interpretations in different representations (consider probabilistic representations of genuinely exogenous risk and of risk generated by internally produced noise or variance);

C. they therefore offer a multitude of explanations of the process of organizational failure or breakdown that are not necessarily commensurable with one another, making it difficult to achieve progress in the study of breakdown through integration across the findings and insights associated with the different models, measures and frameworks.

The (*I-K*) -based approach addresses these problems by accommodating the known representations of organizational complexity in a coherent framework that allows us to make novel, testable predictions about the *rate*, *reparability and reversibility* of organizational failures or breakdowns

First, a few specifications:

a. *Model-Independence.* It is important to observe first that (I,K) framework presents us with a way of representing organizational complexity-driven failures in a way that is not dependent on any *one* or *given* model of organizations (as Boolean networks, or, as epistemically and substantively rational and unbiased processors of information and decision engines). Rather, it is possible to represent any model of organizations within the framework, and obtain a 2-dimensional complexity measure associated with it. It is also, of course, possible to make modifications that lie *within* model classes and retain the ability to use the (*I,K*)-based representation for organizational complexity: one can, for instance, change the basic activity sets that are considered to be critically linked, the nature of these activities and even the definition of what constitutes an activity, while retaining the ability to apply the 2D complexity model to back out a complexity measure;

b. *Model-Invariance*. It is not, however, the case that the complexity *measure* we arrive at will be i*nvariant* across a change of fundamental models: if an organizational task is (I_1, K_1)-complex under representation R_1 and (I_2, K_2)-complex under representation R_2, then there is no guarantee that $I_1=I_2$ and $K_1=K_2$; and that is as it should be, as changes in basic categories and nature of the representation should be expected to effect changes in the fundamental complexity of an organizational task. A *simple* (in the (I,K)-sense), self-fulfilling self-model used by a top manager can be adopted across the organization to produce *simple*, self-fulfilling self-schemas for other managers and *complex* (in the (I,K) sense again) resulting organizational tasks and activities;

c. *Model-Transcendence*. The (I,K)-based representation affords complexity modelers and researchers the opportunity to compare results that link complexity to organizational failure *across* different underlying models of the organization. The language of 'moves in a space spanned by (I,K)' transcends any particular instantiation of an organizational model relative to which complexity was measured and the link between complexity and failure was established.

With these points in mind, we now reconstruct a typology of complexity-driven organizational failure. We distinguish between *K-driven failures* and *I-driven failures* and highlight the mechanisms underlying the two different types of failure are qualitatively different. Organizational *failure* will be considered as an irreversible, systematic, costly and irreparable (or, very costly to repair) breakdown in the predictive capabilities of decision makers about the causal consequences of their actions in the context of the organization and its environment. 'Accidents' and 'mishaps' are distinguished from failures in that they are not *systematic*: what makes an entity 'systematic' is the extent to which its occurrence is causally linked to the basic patterns of behavior of the organization – the extent to which it predictably results from the way the organization behaves, as opposed to being related to exogenous events.

K-Driven Failures. The definition of organizational failure just articulated makes it possible to create a map of organizational failure modes as a function of the mismatch between the *K*-load of a manager's model of the relevant activity of organization - including its response to the environment – and the computational complexity of the most predictively competent ('ideal') model thereof. We make two critical stipulations:

a. Beside the 'simple' or tractable (*P*-hard) and 'hard' or intractable (*NP*-hard) tasks there are also tasks that provably never converge – uncomputable, or 'infinite' ones. Such tasks can model situations in which the 'assumptions' or 'axioms' of a model are not logically compatible: from '*a* and *not-a*', anything at all can be proved [Popper, 1959] and thus a proof procedure running on an axiom set containing both of '*a* and *not-a*' will never converge to a 'steady state'. By the same mechanism, business plans based on internal contradictions will likely be associated with never-ending chains of operations, representing tasks that will never converge to an output;

b. In situations where both the managerial plan and the ideal plan are un-computable, one cannot properly speak of failure properly. One can, however, speak of *breakdown* of the organization, as there is *no* task model that actually converges to an output, corresponding to an endlessly *dithering* organization;

c. The *rate* at which the organization fails is correlated to the size of the mismatch in the K-load class of the managerial and ideal task models. The intuition for this is simple:

I. Tractable (P-hard) models can *approximate* other tractable (P-hard) models, such that errors accumulate only slowly, and are reparable without changing complexity class. This is the typical realm of repeated and cumulative errors;

II. Tractable (P-hard) models can also approximate (although more poorly) intractable (NP-hard) models, but errors accumulate much more rapidly and are not reparable without changing K-complexity class of the managerial model. Failure in such cases is slow to materialize (but accelerates over time);

III. Tractable *or* intractable models cannot *approximate* un-computable models. Failure in such cases is sudden or instantaneous.

With these stipulations in mind, let us examine the kinds of *K*-driven failures that we can now distinguish among, with examples drawn from possible organizational predicaments confronting Hora and Tempus (see Table 3 for a summary):

Complexity Class of Planner's Model of the Task

Complexity Class of 'Ideal' (Competent) Model of the Task	*Tractable (P-hard)*	*Intractable (NP-hard)*	*Impossible (un-computable, or, undecidable)*
Tractable (P-Hard)	No failure. Errors are possible but inexpensively corrigible..	No failure. Errors are possible and more expensive, but still corrigible.	'Debacles': No plan, as minimal computability is a pre-requisite for 'plan' to function as a coordinative device.
Intractable (NP-hard)	Slow failure due to gradual accumulation of incorrigible errors.	No failure. Errors are possible and expensive, but still corrigible.	'Debacles': No plan, as minimal computability is a pre-requisite for 'plan' to function as a coordinative device.
Impossible (undecidable)	Fast failure due to irreversible breakdown in predictive power of model or coordinative power of plan.	Fast failure due to irreversible breakdown in predictive power of model and coordinative power of plan.	'Breakdown': organizational dynamics do not converge.

Table 1.Map of K-Driven Failures.

Slow Failure Through Error Accumulation: P-P Mismatches. When managers' task models and ideal task models are both tractable (P-hard), though mismatched (missing variables or different parameters), we would expect that complexity-driven failure occurs through the accumulation of errors over time, and will be proportional to the size and nature of the mismatch between the managers' task model and the ideal task model. Errors are reparable without changing complexity class, and thus repair costs are low relative to other classes. Suppose the managers of Tempus, Inc. have built a linear demand model and estimated its parameters using standard econometric methods from actual data, but have not properly taken into account the influence of some variables such as seasonal shifts in willingness-to-pay (i.e. their coefficient estimates for these variables are incorrect relative to an ideal or correct model). This mismatch produces errors in demand forecasting, which can accumulate over time and lead to organizational failure (by a massive over-investment in non-revenue-generating inventory, for instance). The eventual failure could at many points in time have been avoided (or, postponed) by the adjustment of the relevant coefficient of the managerial task model. Accordingly, the '*P-P mismatch*' failure regime may be said to be forgiving to managers and their organizations.

Fast Failure Modes. By contrast to the above case, consider two situations that lead to fast K-load-driven organizational failure. In the first, managers' task models are tractable (*P*-hard), but ideal task models are not tractable (*NP*-hard). Suppose managers at Hora, Inc. face a sudden demand up-shift and must suddenly optimize watch production along a large assembly line. They have, however, only a scant understanding of maximal network flow models (which are *NP*-hard) that would allow them to optimize flow of parts and assemblies through the production line, and use simple, linear (*P*-hard) input-output ('first in first out', for instance) models of part and assembly flow through operations when they forecast the rate at which they will be able to fill demand for their products. Over time, errors compound rapidly because of the complexity class mismatch (as the difference between a greater-than-polynomial function and a polynomial function will always be a greater-than-polynomial function) and will lead to a fast failure mode. Moreover, this failure can only be avoided by changing the *K*-load class of the task model that Hora's managers use (from *P* to *NP*), and thus it must be considered relatively unforgiving to the managers and their organization. In the second situation, managers' task models are intractable (*NP*-hard) – as are ideal task models – but they are mismatched (in variables or parameter values) and the complexity difference between the two models is a greater-than-polynomial function of a number of relevant variables that change in time. Once again, failure will be fast and unforgiving. This situation is exemplified by errors and flaws in competitive analysis that employs computationally heavy game-theoretic models of competitors' actions, best responses, beliefs about other competitors, etc. Because iterated dominance reasoning is *NP*-hard [Gilboa, 1989] it is highly sensitive to assumptions about initial conditions and the number and value of variables and therefore highly susceptible to errors in such assumptions [Moldoveanu and Bauer, 2004]. If managers at Hora and Tempus predicate their operational and financial assumptions on profit margins and revenues generated by a duopoly model in which they share one relatively undifferentiated market, but, unbeknownst to them, an entrant (Kayros) suddenly enters the same market, their predictive capacities will break down quickly as a function of 'market time'. Similarly, if, in the duopoly market, Hora's managers assume that Tempus' managers are rational and computationally capable of thinking their way to equilibrium (which Tempus' managers are), but Tempus' managers, while themselves rational, do not believe that Hora's managers are rational and computationally capable of thinking their way through to equilibrium, then, over repeated instantiations of the market and without learning, profit margins will significantly lag behind the optimum achievable levels. Note that, even if it occurs in such instances, managerial and organizational learning will be more costly and difficult than in other situations, because the complexity of the underlying models acts as a *noise amplifier* [Moldoveanu and Bauer, 2004] which increases the variance of testable predictions and hypotheses about competitive interactions.

Instantaneous Failure Modes. Instantaneous failure modes will obtain in mismatches in which at least one of the two task models (managerial, ideal) will be uncomputable (or, will generate an undecidable problem). Consider, for instance, a situation in which Hora's managers come to use two different language systems to refer to observable events (assembly line failures of a particular kind) that they – on reflection – would hold to be *indistinguishable under all measurable aspects*, and evolve adequate but mutually incompatible organizational routines for 'handling' these events, which are causally connected with the two language systems in question. If they are unaware of this ambiguity or – even though aware of it – have not evolved a successful way of playing the coordination game such that they 'fall into' one or another routine set (but not both), then the organization as a whole may be said to be 'stuck' in an uncomputable ideal task model and will fail *suddenly* or *instantaneously*, no matter what the managerial model of the situation may be (tractable, intractable or uncomputable).

Uncomputability and instantaneous failure need not result from error. They can arise from technological or demand discontinuity that amounts to a paradigm shift in the Kuhnian sense [Kuhn, 1962]. If Kayros (the entrant) succeeds in seeding and growing a demand for 'time measurement and management experiences' (enabled by a centrally located software agent that targets the end user on any meridian or parallel with precise time estimates and up-to-date relevant news which it downloads to any one of several electronically linked devices) rather than 'watches' to the point of challenging the Hora-Tempus duopoly and threatening to replace 'time measurement instruments' with 'time measurement experiences' , the problem of *deciding* which of the two end products will 'win out' in the end will be undecidable either in the 'objects framework' of Hora and Tempus (where only durable goods exist and are exchanged) or in the 'events' framework of Kyros (where only experiences and signals exist and are exchanged).

The same kind of failure will obtain if the managerial task model is uncomputable (for instance, if it contains logical inconsistencies), even if the ideal task model is itself computable (tractable or intractable). The most intuitive example for this kind of failure comes from examining the effects of failures of logical closure in a business plan, which we can do if we consider a business plan as a logical model (with 'assumptions', 'theorems' and 'hypotheses' that are tested by 'data'). If Hora's business plan calls for particular levels of investment during the following two quarters to support the development of a new watch design on the basis of assumptions about the demand for new designs and trailing earnings before income taxes and depreciation and current gross margins are hit by a price war with Tempus while design requirements have undergone subtle changes, then the organization will fail in K-mismatch sense we have outlined here if its managers do not work the logical implications of these new conditions into their business model. Of course, working through all of the logical implications of a change in conditions may be an intractable task (*NP*-hard), and engaging in it in such a situation can take the organization from one mode of possible failure (instantaneous on account of uncomputability) to another (fast on account of rapid error propagation).

I-Driven Failures. To understand the kind of failure informational depth can lead to, consider the problem of predicting the evolution of a process whose most compact representation takes up L bits of memory. If a manager's model of the task has informational depth of $L-1$ bits, then he or she will *never do better than chance* in predicting the evolution of the process in question. Even if the manager gets all of the $L-1$ bits 'right', there will be an extra non-redundant bit in the valid description of the process in question that the managerial model has no way of predicting. There exists no *2*-bit representation of the number *9*. The constraint on the informational depth of the managerial model may be 'hard' – it may dictated by the ultimate beholding capacity of the manager or the organization as a whole – or it may be 'soft', in which case the manager can change the complexity class of his or her model and therefore hope to change complexity class of his or her model and thus match the fundamental informational depth of the process in question.

Following this intuition, we stipulate - as we did in the *K*-driven failure case – that there exists a managerial task model (represented by a program) of informational depth I_M and an 'ideal task model' represented by a program of informational depth I_I - but we add – in each case – a finite *memory* meant to *hold* the task models, corresponding to the *beholding* capacity of the manager or of the organization as a whole. (It is not required for a single managerial mind to behold the model in question: *I* can be distributed among many members of the organization in ways that are or are not – as in Weick's 'group mind' examples [Weick, 2005] – reducible to a linear aggregate of the models beheld by individual managers.)We now compare to the size of the 'working memory' the informational depth of the ideal model of a task, process or phenomenon (the 'task model') in order to distinguish between the following regimes:

A. the **sub-critical** regime: the size of the minimal program representing the ideal task model is lower than the size of the memory available to represent it;

C. the **super-critical** regime: the size of the minimal program representing the task model is greater than the size of the memory available to represent it.

We can now make useful distinctions among different kinds of *I* –driven organizational failure by considering mismatches between the informational depth of managerial and ideal task models and their implications for the likelihood and reparability of failures of predictability. We will assume that the ideal task model will in all cases have higher informational depth than the managerial task model and ask: what kind of failures obtain in the different regimes, Table 2):

	I- Complexity Class of Planner's Model of the Task and Actual Plan.		
***I*-Complexity Class of Competent Model of the Task and 'Ideal' Plan.**	*Subcritical*	*Critical*	*Super-critical*
Sub-critical	No failure. Errors are possible but inexpensively correctable.	Infrequent failure. Errors are possible but inexpensively correctable.	'Conundra': no plan is feasible.
Critical	Fast failure. Organizational response to plan is fundamentally 'unpredictable'.	Infrequent failure. Errors are possible and more expensive, but corrigible, with the exception of errors arising from non-self-delimiting programs.	'Conundra': no plan is feasible.
Super-Critical	'Breakdown': preconditions for existence of organization are negated in this regime.	'Breakdown': preconditions for existence of organization are negated in this regime.	'Breakdown': preconditions for existence of organization are negated in this regime.

Table 4. I-Driven Failures.

The Sub-Critical Regime. If the ideal task model can in theory be beheld by the organization, then *I*-driven failure is unlikely: errors caused by mismatches between I_I and I_M can be *rectified* by systematically searching through the bit strings up to the beholding capacity M of the organization (there are $\sum_{k=1}^{M} 2^k - M$ of them). If Hora's managers have a competitive model that does not include Tempus, Inc.'s managers' correct conjectures about Hora, Inc.'s managers' conjectures about the market, but their model of competitive interactions *does* have the additional degrees of freedom corresponding to such conjectures, then their model can be rectified to include the new and relevant information. Note that in this case the organization can still fail from the accumulation of errors over time (slow or fast K-driven failure) arising from the accumulation of errors or incorrect estimates of the computational load of the task model.

The Super-Critical Regime. If the ideal task model has an informational depth that exceeds that of the beholding capacity of the organization, the organization will *fail in the I-sense* with probability that is proportional to the mismatch between the informational depths of the two models. A two-bit prediction of an event that takes a minimum of 3 bits to represent will be right ½ of the time (assume the third bit will be random, as it lies above the representational capacity of the model); it will be correct ¼ of the time when it attempts to predict events that take a minimum of 4 bits to represent, and, in general, it will be right with probability 2^{2-I_e} when attempting to make predictions about events whose representations have informational depth I_e. If, for instance, the competitive model that Hora, Inc's managers use has no *conceptual space* for Tempus' conjectures about Hora's strategies and payoffs – heeding these variables simply 'does not cross their minds' - - and these conjectures are relevant to the outcome of the competitive interaction, then Hora, Inc. will achieve sub-optimal outcomes because of an I-driven failure that they cannot rectify by changing their model. The super-critical regime we have identified can be understood as capturing organizational failures driven by 'unknown unknowns' – which lie outside of the scope of managerial awareness because of informational depth constraints on the complexity of the organizational or managerial model.

5. Concluding Comments. We have characterized organizational complexity using a 2 dimensional measure that captures two essential and independent characteristics of the difficulties associated with coping with complexity. We called these dimensions informational depth and computational load and showed how a two-dimensional complexity measure allows us to understand organizational complexity coping mechanisms and organizational failure modes and regimes.

The new complexity measure allows us to make explicit and clarify assumptions that are embedded in existing models, to make new and useful distinctions regarding the representation of organizational phenomena, and to open up new research questions and programs of inquiry.

a. *Explicitation of assumptions*. We showed that existing approaches to complexity rely on *a priori* models or representations of organizational phenomena for the definition of complexity. Definitions of complexity as a degree of linkage or coupling among different components of the organization, for instance (dating back to [Simon, 1962]), rely on specific partitionings of organizations into 'parts and connections' (or, nodes and edges, or, entities and links); they are not invariant across different possible representations of the organizational whole. Definitions of complexity in computational terms [Rivkin, 2000, for instance] have thus far relied on specific underlying models (NK(C) models in this case) that make possible the measurement of the computational complexity of making predictions about their temporal evolution. By explicitly introducing the additional step of modeling the process of modeling itself and measuring the complexity of the resulting model on a computational device, we have made the dependence of the complexity of a phenomenon on the model used to represent that phenomenon explicit. This move makes it possible to speak of the complexity of organizational phenomena modeled on the basis of different conceptual schemata. The complexity measure that we have introduced is still dependent on the choice of *language* in which models are articulated, or, equivalently, on the choice of Turing Machine used to simulate the phenomenon in question. This is a fundamental dependence, inescapable in view of the language-dependence of any observational or theoretical statement [see Kuhn, 1990, for instance]. Our framework allows us to make this dependence explicit.

b. *New Distinctions*. We can now usefully distinguish between an informational and a computational component of the complexity of an organizational phenomenon, which allows us a more precise characterization of the *organizational costs of complexity* and permits us to link informational depth and computational load explicitly to costly organizational processes (beholding and executing) on which the organization as a whole may be understood to try to economize. Accordingly, organizational adaptations to complexity may be categorized using the new framework as moves and maneuvers that attempt to economize on I alone, K alone, moves along the I-K frontier of the organization and shifts of the frontier itself. Moreover, complexity-driven organizational failure modes driven by I and K were distinguished from each other, making an inquiry into the complexity-driven failure of organizations more precise.

New Questions. The new framework makes it possible to ask new questions about organizational adaptations to complexity that investigate the structure and dynamics of organizational complexity:

I. *What is the I-K complexity profile for various kinds of organizational production functions and does this profile correlate with organizational profitability?* If a fundamental 'complexity capability' exists that allows organizations (such as IBM, Intel, Microsoft) to consistently register above-normal returns by executing tasks that lie above a complexity frontier that bounds the activities of other firms, it is now possible to analyze this capability and explore the 'complexity profile' of organizations in terms of *I* and *K*.. It is also possible to make predictions about the activity sets that organizations can tackle to leverage such a complexity capability. A move by IBM into bio-informatics looks like 'diversification' under the usual definitions of organizational capabilities, but appears to be consistent with the complexity profile of the organization.

II. *Do I or K bounds to organizational complexity serve as leading predictors of organizational failure?* By making measurements of the complexity of production tasks possible, it is possible to make predictions about the likelihood of incumbents to survive technological or demand paradigm shifts (which can produce step function increases in both *I* and *K*.

III. *What is the complexity of the (organizational learning) processes?* We have shown it is possible to measure the computational load of search processes and the informational depth required to conduct them. By combining these two costs, it is possible to make predictions about the true organizational costs of learning and to pose the organizational learning problem as an *optimization exercise in complexity space*. This approach can be extended to the question regarding the complexity of learning about complexity, which would allow us to understand, in complexity space, the complexity economics of self-understanding.

7. **References.**

Abelson, H., G.J. Sussman, and J. Sussman, (1984): The Structure and Interpretation of Computer Programs; Cambridge, MA.: MIT Press.

Anderson, P.W., (1999) *Complexity theory and organization science*, Organization Science, 10:3.

Baldwin, Carliss Y. and Clark, Kim B. (2000): Design Rules; Cambridge, Mass.: MIT Press.

Barnsley, Michael F. (1988): Fractals Everywhere; San Diego, CA: Academic Press.

Barnsley, M.F. (1986) Fractals Everywhere, New York: Academic Press.

Bennett, Charles H. (1988): *Logical Depth and Physical Complexity*; in: Herken 1988, pp. 227-257.

Berlinski, David (2000): The Advent of the Algorithm; San Diego - New York - London: Harcourt.

Brown, Shona and Eisenhardt, Kathleen M. (1998): Competing on the Edge; Boston, Mass.: Harvard Business School Press.

Casti, J. (1991), Reality Rules: Picturing the World in Mathematics, New York: John Wiley.

Chaitin, Gregory J. (1974): *Information-theoretic Computational Complexity*; IEEE Transactions on Information Theory, 20, pp. 10-15.

Chaitin, Gregory J. (1982): *Gödel's Theorem and Information*; International Journal of Theretical Physics, 22, pp. 941-954.

Chapman, William L., Rozenblit, Jerzy and Bahill, A. Terry (2001): *Systems Design is an NP-Complete Problem*; Systems Engineering, 4/3, pp. 222-229.

Church, Alonzo (1936): *An Unsolvable Problem of Elementary Number Theory*; American Journal of Mathematics, 58, pp. 345-363.

Cohen, M.D. (1999), *Commentary on* Organization Science *Special Issue on Complexity*, organization Science, Vol. 10:pp373-376.

Cormen, J., C. Leiserson, and R. Rivest, (1993): Introduction to Algorithms; Cambridge, MA: MIT Press.

Cover, Thomas M. and Thomas, Joy A. (1991): Elements of Information Theory; New York et al.: Wiley & Sons.

Cyert, Richard M. and March, James G. (1963): A Behavioral Theory of the Firm; Englewood-Cliffs: Prentice Hall.

Ethiraj, Sendil K. and Levinthal, Daniel (2004): *Modularity and Innovation in Complex Systems*; Management Science, 50/2, pp. 159-173.

Garey, M.R. and Johnson, D.S. (1979): Computers and Intractability: A Guide to the Theory of Incompleteness; New York: Freeman.

Gilboa, I (1989*): Iterated Dominance: Some Complexity Considerations*; Games ans Economic Bahvior, 1, pp. 15-23.

Goodman, Nelson (1978): Ways of Worldmaking; Indianapolis - Cambridge: Hackett; dt: Weisen der Welterzeugung, Frankfurt a. M.: Suhrkamp 1990.

Holland, John H. (1975): *Adaption in Natural and Artificial Systems*; Cambridge, MA: MIT Press.

Holland, John H., Holyoak, Keith J., Nisbett, Richard E. and Thagard, Paul A. (1986): Induction: Process of Inference, Learning, and Discovery; Cambridge: MIT Press.

Kauffman, Stewart A. (1988): The Evolution of Economic Webs; in: Anderson / Arrow / Pines 1988, pp. 125-146.

Kauffman, Stewart A. (1993): The Origins of Order; New York: Oxford.

Kuhn, T.S. (1962) The Structure of Scientific Revolutions, Chicago: University of Chicago Press.

Kuhn, T.S. (1990**),** The Road Since *Structure*, Cambridge: MIT Press.

Levinthal, Daniel A. and Warglien, Massimo (1999): *Landscape Design: Designing for Local Action in Complex Worlds*; Organization Science, 10, pp. 342-357.

Lewis, D. (1969) Convention: A Philosophical Study, Cambridge: Harvard University Press.

Lewis, Marianne W. (2000): *Exploring Paradox: Towards A More Comprehensive Guide*; Academy of Management Review, 25/4.

Li, M. and P. Vitányi, Paul (1997): An Introduction to Kolmogorov Complexity; New York: Springer; 2nd edition.

March, J.G. and J. Olsen, (1972), Ambiguity and Choice in Organizations, New Jersey: Prentice Hall.

March, J.G. and H.A. Simon. (1958): Organizations; New York: Wiley.

Marengo, L, G. Dosi, P. Legrenzi and C. Pasquali, (2000), *The Structure of Problem Solving Knowledge and the Structure of Organizations*, Industrial and Corporate Change, 9:757-788.

McKelvey, B., (1999), *Avoiding Complexity Catastrophe in Coevolutionary Pockets: Strategies for Rugged Landscapes,* Organization Science, Vol. 10: pp. 294-321.

McKelvey, Bill (2004): *Toward a complexity science of entrepreneurship*; Journal of Business Venturing, 19/3, pp. 313-341.

Meltzer, B. and Michie, D.,Eds. (1969*): Machine Intelligence* 5; Edinburgh: Edinburgh University Press.

Miles, Raymond E. and Snow, Charles C. (1978): Organizational Strategy, Structure and Process; New York - Tokyo - Hamburg: McGraw-Hill; dt.: Unternehmensstrategien, New York - Tokyo - Hamburg: McGraw-Hill 1986.

Miller, D. (1993), *The Architecture of Simplicity*, Academy of Management Review, Vol 34: 21-39.

Miller, Randolph A., Pople Jr., Harry E. and Myers, Jack D. (1982): *INTERNIST: An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine*; New England Journal of Medicine, 307, pp. 468-476.

Mintzberg, Henry (1979): The Structuring of Organizations; Englewood Cliffs: Prentice-Hall.

Moldoveanu, Mihnea C. and Bauer, Robert (2004): *On the Relationship Between Organizational Complexity and Organizational Structuration*; Organization Science, 15/1, pp. 98-118.

Nelson, R. and S.G. Winter, (1982), An Evolutionary Theory of Economic Change, Cambridge: Harvard University Press.

Ortmann, Günther (2003): Regel und Ausnahme; Frankfurt a.M.: Suhrkamp.

Papadimitriou, Christos H. (1994): Computational Complexity; Reading, MA et al.: Addison-Wesley.

Perrow, Charles (1970): Organizational Analysis: A Sociological View; London: Tavistock.

Poole, Marshall Scott and Van de Ven, Andrew H. (1989): Using Paradox to Build Management and Organization Theories; Academy of Management Review, 14/4.

Popper, K.R. (1959), The Logic of Scientific Discovery, London : Unwyn Hyman.

Quinn, R. E. / Cameron, K., (Eds.) (1988): Paradox and Transformation: Towards a Theory of Change in Organization and Management; Cambridge, MA.: MIT Press.

Rivkin, Jan (2000*): Imitation of Complex Strategies*; Management Science, 46, pp. 824-844.

Scott, R. (1985), Organizations: Rational, Open Systems, Beverly Hills: Sage.

Shannon, C.E. and J. Weaver (1949), The Mathematical Theory of Communications, New Jersey: Prentice Hall.

Simon, Herbert A. (1962): *The Architecture of Complexity*; Proceedings of the American Philosophical Society, 106, pp. 467-482.

Thiétart, Raymond-Alain and Forgues, Bernard (1995): *Chaos Theory and Organization*; Organization Science, 6: pp. 19-31.

Turing, Allen M. (1936): *On Computable Numbers, with an Application to the Entscheidungsproblem*; Proceedings of the London Mathematical Society, 2/42, pp. 230-265.

Turing, Allen M. (1948): *Intelligent Machinery*; National Physical Laboratory Report, Wiederabdruck in: Meltzer/Michie 1969, pp. 3-23.

Weick, K.E. (1995), <u>Sensemaking in organizations,</u> Beverly Hills: Sage.

Weick, K.E. (2005), <u>Making Sense of the Organization</u>, Beverly Hills: Sage.

Westhoff, Frank H., Yarbrough, Beth V. and Yarbrough, Robert M. (1996): *Complexity, Organization, and Stuart Kauffman's "The Origins of Order"*; Journal of Economic Behavior and Organization, 29, pp. 1-25.

Wittgenstein, L. (1953), <u>Philosophical Investigations</u>, New York: Cambridge University Press.