Managing Dynamic Complexity
A Biomimetic Perspective

Caryl Erin Johnson, PhD
Chief Innovation Officer
Introspective Systems LLC

Introduction

I would like to share with you today some thoughts that I have been nurturing for most of my professional career as a professor, a civil servant, and in corporate applied science. As to the latter, I consider myself to be an applied scientist, which I propose to be somewhere in between a basic research scientist and a classical engineer. At BAE I was a systems architect. There is a certain class of problem which seems to be pervasive across many domains of endeavor, though it was not until quite recently that I have been able to put a name to it. Also, I have been interested in INCOSE for several years, especially when I first learned of the organization when its focus was more directed at fractal systems of systems, although that emphasis seems to have diminished somewhat in the last few years. But the story I wish to tell is one of complexity and how to interact with it.

Managing Dynamic Complexity

Complexity is one of those words that people everybody knows, but in many cases do not know in quite the same way as each other. At its most fundamental meaning, complexity suggests something that is so complicated that it is difficult to understand. However, when scientists discuss the field of 'complexity theory' they are referring to something much more precise, but perhaps just as difficult for the lay person to understand. In this paper and in the applications described later, we are focusing on the latter specifically in the context of complex adaptive systems.

Peter Senge, in his book 'The Fifth Discipline' distinguishes between two kinds of complexity, Detail Complexity and Dynamic Complexity. Detail complexity is more aligned with the lay understanding that complexity refers to complicated things in that they have a lot of components or variables, two many to be understood in a holistic manner. However, when we think of Complex Adaptive Systems, we are thinking of Dynamic Complexity which Peter Senge defines as:

> *…situations where cause and effect are subtle and where the effects over time of interventions are not obvious. Conventional forecasting, planning and analysis methods are not equipped to deal with dynamic complexity.*

As conventional analytical methods are inadequate to address dynamic complexity, the remainder of this presentation describes a biomimetic approach using a fractal executable graph (xGraph ™) to manage dynamic complexity.

Three Pillars of Dynamic Complexity

In our experience, as we consider the problems that seem to repel solution, three factors emerge that in many respects appear to define dynamic complexity. We identify these three factors as 1) systems having many diverse actors including both cyber and human, 2) a goal to seek an optimal distribution of scarce resources in pursuit of a generally well-defined objective, and 3) systems having an extremely high dimension and generally non-linear decision space. In most cases we attempt to model and manage such systems in the form of a Markov Decision Process, or MDP, although in truth this characterization does not technically apply. In an MDP, a system is understood to be in some state after which an action is chosen, and the system transitions to another state, although in stating this we admit to a certain amount of over simplification. In such systems, the Markov condition applies, which is the outcome of the decision is only dependent on the state and not how one achieved that state.

Unfortunately, for dynamic complex systems this definition does not strictly apply. Generally, we think of an MDP as having a single actor making a decision, while in a dynamic complex system there are many actors. For example, in even the simplest natural ecosystem, which is a prime example of dynamic complexity, the number of actors making independent decision surpasses trillions. The problem is further complicated due to the fact, that each of these possibly trillions of actors has an incomplete awareness of the state, as when a frog does not know that an owl is about to select it as its next meal. In addition, even if each actor did have a complete knowledge of the current state of the system, something that is clearly an impossibility in a global sense, the outcome of any decision is entirely stochastic. Apparently, one might think of dynamic complexity as at best a Partially Observed Markov Decision Process (POMDP) with multiple actors. In such systems, again such as an ecosystem, dynamic stability is achieved by positive feed back loops between the various actors, something that is often represented as a 'food chain'. In a Dynamic Complex System solution, if you can call them that, results are emergent in the sense of a self-organizing characteristic of such systems.

Swamp: Natural Dynamic Complexity

In choosing a swamp as a naturally occurring example of dynamic complexity, we are not trying to achieve any validity by invoking the concept biomimicry. Our only intention here is to use it as an example, and we believe an excellent example, to develop a better understanding of how the management of such systems might be achieved. At the very least, we hope to build a better understanding of how to identify such systems in the 'wicked problems' that we confront in our lives, studies, and occupations. Also, there is a rich body of knowledge regarding how positive feedback in an ecosystem influences the evolution of such systems.

In the swamp, the actors are the various and diverse creatures big and small that inhabit this environment. Each of these actors, including the evolutionary learning manifest in vegetation's dissemination of seeds, is independent, yet at the same time linked into a complex web of interdependency. In reference to the second pillar of dynamic complexity, the resources in a swamp, such as light, space, nutrients, and such, are strictly limited, and dynamic stability is achieved in a kind of competitive interplay amongst the various actors. At the same time, in reference to the third pillar, each actor has many choices, as in a frog can chose to jump, sit still, eat a bug, or seek a mate, but regardless of what action is selected, the outcome is not necessarily what might have been intended. In fact, this is what is manifest in our management problem, in that as situations increase in dynamic complexity, as in an expanding business, the best of intentions resulting from any decision might be defeated by the negative effect of unintended consequences. This is one reason why I personally have steadfastly avoided the trap of upper level management positions – I still enjoy sleeping at night.

The Elephant in the Living Room

Possibly the first step to managing any problem is to see that it exists. While this might at first seem trite, in the case of confronting dynamic complexity, our experience is that this is quite challenging, hence the metaphor of the elephant in the living room. There are very few of us that would not characterize our lives as complex, comprising many interactions and decisions that we do not fully understand. At first, we might see this as simply detail or static complexity, and in many cases it is, but as one's scope of responsibility increases, such as in advancement in corporate management, static complexity slowly evolves into dynamic complexity, yet we might still view it as just a complicated situation. In fact, in developing our customer relationships, at Introspective Systems, we experience this as one of our greatest challenges. Perhaps part of the reasons for this is that as humans we tend to be more cognizant of things that are changing more quickly, while remaining unaware of situations that evolve at a much slower pace. In any case, methodologies that have proven effective in addressing simpler manifestations of complexity slowly become less effective.

Another factor that confounds cognition is that it hard to see something that is all around us. For example, how many people think about the fact that we survive only because we are embedded in an atmosphere that is comprised of 21% oxygen – we simply take breathing for granted. Does a fish think about alternatives to water – I doubt it. There is no concept of 'not water' to use as a bases for thinking about water as anything other than just what is. Similarly, for dynamic complexity, we tend to be so embedded in it, that it ceases to have any direct meaning, and hence our first challenging in developing a solution for a new client is to find a way for understanding and sharing that understanding of how dynamic complexity is defeating effective decision management in their endeavors. That is the very first step on the path to managing complexity.

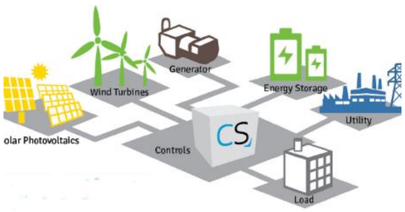
Understanding the State Space

After overcoming the cognitive challenging of seeing that an organization's confounding problem is characterized as one of dynamic complexity, the very next hurdle in managing such problems is how to achieve collaborative decision making in a context where every dactor sees the problem in a different light – hence the second elephant metaphor. Communication, that is the interactions between the actors must play a key role, but even so, the outcome can be more accurately characterized as thinking globally but acting locally. In principal every departmental VP shares the goal of helping an enterprise thrive, but each contributes to that goal in a different way, and with a different and more parochial vision of the state space. Again – think globally and act locally, and somehow that process must be conducted in a holistic manner.

Transcending the Cognitive Barrier

But now we get down to the crux of it, or in the vernacular, where the rubber hits the road. This is the challenge that Introspective System faces, and frankly we have not been able to come up with a satisfying approach. By transcending the cognitive barrier, I am referring to finding a way to communicate to potential usesr of a technology a vision of how that technology can solve a problem that they do not in fact know that they have. This is what I was referring to earlier as the elephant in the living room. While I cannot suggest a clear solution to this dilemma, I can in the next three slides discuss how the issue of dynamic complexity arises in three specific domains.

Example: Microgrid



Much of our experience and research has come from working with the US Department of Energy on a national grid architecture that decentralizes control. What this means is that rather having all of the information flow back to a central control point, most of the decisions regarding energy distribution is made at the edge, that is near or at the generation or energy consumption points. An example of centralized control is something called demand side management (DSM), where message are sent from a central facility to start and/or suspend generation and load in response to instantaneous demand. With edge control, these decisions are made by devices at the edge, and in the research that we have been conducting, this is accomplished by pricing signals that can be used by edge devices in the sense of buy or not buy decisions. As a specific example, an energy storage device can learn a strategy wherein it buys energy and stores it when energy is cheap, and then sells it back when the price is high.

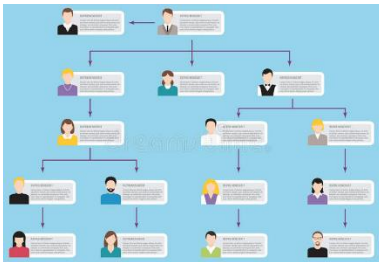
This is a clear example of dynamic complexity. There are many actors of many different types, power is a scarce or at least limited resource, and the decision space is extremely high. However, and the reason that this is the first example, is that it is something simpler than others, because the behavior of the devices is for the most part, barring failures, predictable. In control theory the statement could be made that each device has a specific plant model – if a switch is closed, a motor will turn on. Yet even in this simple example, the question of whether a large-scale distribution of such devices can achieve dynamic stability in the same way as our natural ecosystem example of a swamp.

Example: StarCraft II.

Our next example is taken from the domain of virtual, multi-player online gaming. Recently, Google working with Blizzard Entertainment has developed an application programming interface (API) that can be used to as a laboratory for developing algorithms for collaborative artificial intelligence. Google puts this forward as the next big challenge in AI development following recent successes in the game of Go, Chess, and others. Indeed it is, because unlike the others, this challenge is one of dynamic complexity while the predecessors only manifest static complexity. In Chess and Go, when you make a move, your can be pretty sure that the piece goes where you put it and stays there – there is no stochastic or nonlinear behavior.

While most users of the StarCraft II API assume a single player and a single opponent, that is two actors, we have morphed the interface so that each of the game pieces is controlled individually by adaptive AI algorithms so that achieving a successful outcome requires collaboration and communication. There is no way to predict the behavior or response of the opposing units, and the game state is at best only partially known. This is dynamic complexity with many actors, limited resources, and an extremely high and stochastic action space. Some of the actors have plant models, but for opposing forces, no plant model is available so responses cannot be predicted but must be learned.

Example: Corporate Management



As a final and third example we are using the challenge of managing a corporation or large enterprise. We submit that every manager at every level (the actors in this case) are facing the challenge of dynamic complexity. There are many and varied actors, each actor is faced with limited resources to enable their decisions, and the size of the action space is vast. Further, none of the actors have plant models, as human behavior is notoriously difficult to predict. To make things even worse, in a large company, information at all level is limited despite the existence of moderately successful management information systems and is sometimes massaged so as to make it more positive than might be the case. It is a classical, multi-actor POMDP and obviously a problem of managing dynamic complexity of the worst kind.

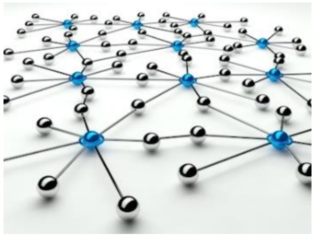
So, every manager is swimming in an ocean of dynamic complexity, and is as unaware that this is what is limiting their effectiveness - just as the fish is unaware of the water. Lest I make the impression that the situation is hopeless, the rest of this talk focuses on how to try to make the best of it.

Managing Dynamic Complexity

So, now to answer the question of how to best manage dynamic complexity. Well, the answer, unfortunately, is that you can't. If you could, it would not be dynamic complexity, but that does not mean that there are not ways to deal with it, and that is exactly what we presume to discuss next, because deal with it as a manager, engineer, scientist, or academician, you must!

While for obvious reasons it is impossible to control or manage any system for which you have incomplete awareness, it is possible to expand that awareness so as to better anticipate the effect of a specific decision or action and thus reduce the likelihood of the unintended consequences that can turn the best of intentions into a negative result. The bottom line is that although you cannot control dynamic complexity, with the proper tools one can learn to influence the outcomes and thus make decisions that are most likely to lead to positive results. As we face decisions of ever increasing complexity, it is inevitable that some of the outcomes are beneficial while others are quite the opposite. For example, in a supply chain, do we prefer one source over another. In investing in a company what are favorable terms or when do we walk away. In the medical industry, what are the most effective diagnostics to schedule to maximize value to the patient while at the same time minimizing unnecessary cost. There are many other such examples. The goal is to nudge this balance so that the favorable outcomes are increased and the negative are decreased. Introspective Systems has developed a tool called xGraph to aid in this process.

xGraph: Executable Graph Framework



The xGraph platform is an executable graph framework that can be applied to a system that manifests dynamic complexity in a manner that allows deeper insight into the responses that might be entailed by any actions taken. The architecture is a fractal graph, modeling a system of systems, where the nodes represent specific actors in the environment. For the microgrid example these would be such things as batteries, solar arrays, wind farms, and loads. Each node or devices is capable of adaptive behavior, which is adjusted as it learns how to optimize a global goal by acting within the scope of locally available information. That is – act locally, but think globally. Collaboration is achieved by nodes communicating with other nodes and exchanging knowledge of the environment. A simulation replicating the behavior of an actual complex system an provide 'what if' scenarios and exploration of high risk parts of the decision space which cannot be attempted in the actualization of the system where mistakes or bad decision are much more costly and possibly catastrophic. There can also be a merging of the actualization and simulation such that the effect of incremental changes can be evaluated before actual deployment. This is sometimes referred to as 'hardware in the loop'. Alan Kay used the phrase 'homoiconic' to describe this relationship between systems and representations of system.

In a sense, xGraph could be mistaken as a highly scalable, self-organizing graph database where the nodes contain both information as well as learned behavior. However, it is most definitely not a graph database because it is entirely self-organizing, and operates without external programs required to manipulate structure or store information. In a manner of speaking, it is more like an organism that consumes information and grows as information is organized into its internal structures as in adaptive learning. It can be observed and even influenced, but it can never be completely controlled.

The purpose of this talk is not to describe in detail how xGraph works or to convince you to consider using one. The point is only to submit that management of dynamic complexity is possible, even as complete control is not. I would certainly be willing to describe the xGraph framework in greater detail if people are interested.

Conclusions

In conclusion, we believe that problems manifesting Dynamic Complexity limit growth and success in some of the most important of human endeavor. Example include response to disaster, managing climate change, corporate management, the national economy, immigration policy, cyber security, or universal healthcare to name just a few of many, many examples. In some domains these problems are called "wicked problems", and they certainly are that. They are found in almost every domain of human activity. To move forward and achieve success it is imperative that we find and develop means of understanding and to whatever extent possible managing these wicked problems, and together we can evolve ever better methods for managing dynamic complexity in our various endeavors. Thank you and I am looking forward to your questions and especially your thoughts and suggestions.