INTROSPECTIVE SYSTEMS

A Practical Approach to the Management of Dynamic Complex Systems

Dr. Caryl Johnson
November 2018

ABSTRACT

Mankind is increasingly faced with the challenges of making effective decisions in complex environments where interacting systems and relevant information evolve even as those decisions are being made. The dynamic complexity of these environments can lead to unintended consequences that undermine the positive outcomes, sometimes to the extent that forward progress becomes impossible. Examples are easy to find in such diverse fields as economics, politics, the environment, corporate management, and investment decisions, to name just a few. While the concept of managing dynamic complexity is in itself a bit of an oxymoron, we developed a self-evolving (AI based) dynamical framework that enables more effective decision making in the face of dynamic complexity.

The solution presented here is called xGraph, an executable graph framework that presents an approach to representing and taming dynamic complexity. This approach allows both the simulation as well as the control of systems within the dynamic environment, a process that is similar to the balance of collaboration and competition in a natural ecosystem. To date, this introspective and reflective technology has been applied in the fields of global seismology, bioinformatics, swarms of autonomous entities, strategic gaming, and the creation of a modern fractal architecture for the national electrical grid.

1: Managing Dynamic Complexity

Complexity is one of those words that everybody knows, but not 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, and perhaps more difficult for the layperson 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.

Senge (2010) described two forms of complexity: 1) Detail Complexity and 2) Dynamic Complexity. Detail complexity is more aligned with the lay understanding that complexity refers to complicated things with a lot of components or variables, too many to be understood in a holistic manner. However, when we think of complex adaptive systems, we are thinking of Dynamic Complexity, which Senge (2010) 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.

To frame the problem and the dire need for innovative solutions, it is important to understand that dynamic complexity confounds almost every area of human endeavor. From corporate executive management, to managing a portfolio of investments, to developing a new drug, to conducting warfare operations, to trying to raise a teenager, dynamic complexity becomes apparent when the balance of good decisions compared to bad decisions tips in favor of the latter. As a practical matter, there ar a dearth of tools and methods to help us think about and understand dynamic complex systems. To make things worse, most of us are unaware of the true cause of our woes.

Coupled with this in the real world is the fact that dynamic systems rarely respond to our actions in exactly the manner that we intend. The understanding that the resistance to progress can be usefully framed in terms of dynamic complexity is generally lacking. Instead, alternatives involving human factors or social limitations often replace reasoned analysis in this assessment, and without a clear framing of the core problem, a solution is impossible. Because conventional wisdom lacks the framework to think about dynamic complexity, we instead think about the individual components of the complex system and try to fix individual components. The effect of this is a form of “the elephant in the room” where something is such a large issue that we don’t want to think about it or talk about it. Instead, to properly address dynamic complexity problems, we need to think about the system as a whole.

There are four characteristics to consider in the assessment of whether a dynamic complexity framework might be useful in describing the phenomenon:

1. The system is comprised of many different actors interacting with each other in diverse manners.
2. There is an extremely high-dimensional decision space of actions that can influence the situation.
3. There is a clear goal or desired outcome that represents success to the extent that it is achieved.
4. There are limited resources to deal with the problem such as information, time, or money.

In “The Tragedy of the Commons”, Hardin (1968) describes the difficulty of managing a shared resource when individuals strive to maximize their own benefit without regard for the common good, ultimately leading to the destruction of the shared resource (and, therefore, individual detriment as well). Hardin opines that this dynamic complex systems problem cannot be solved by technical means. However, the xGraph framework is a technical solution designed to address dynamic complexity.

The following sections describe natural and synthetic ecosystems in more detail, and present the xGraph framework, which is designed to address dynamical complexity problems. Section 2 describes natural ecosystems; Section 3 describes computational ecosystems; Section 4 describes the specific xGraph computational framework; and finally, Section 5 presents some example xGraph projects. The basic premise is that a practical solution can be framed in terms of a POMDP (Partially Observed Markov Decision Process) which can be simulated using homoiconic (Kay, 1969) modeling in an executable graph framework (xGraph) leading to the understanding that can allow an advance in effective decision making.

2: A NATURAL ECOSYSTEM

Sometimes the solution to some of our hardest problems can be found in nature. As described by Vincent et al. (2006), “biomimetics” entails “the practical use of mechanisms and functions of biological science in engineering, design, chemistry, electronics, and so on.” In other words, “biomimetics” refers to using natural systems as inspirations for engineered systems.

As an example of an easily scalable ecosystem, consider a swamp. A swamp comprises a very diverse array of 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. A swamp manifests extremely limited resources in terms of light and nutrients and dynamic stability is achieved in a both a cooperative as well as competitive interplay amongst the various actors. This understanding will be critically important in the discussion of the computational analog, where it is seen that collaboration in a swarm of autonomous entities requires both cooperation and competition as essential components of collaboration. While one might imagine that at any instant of time, a swamp might be seen to be in a specific state in reference to a POMDP, in actuality the individual creatures have a limited and localized notion of state space that is relevant to their survival. For example, the decisions made by a snake to capture and eat a frog is quite different than that of a frog whose main interest is eating flies while avoiding being eaten by a snake. In both cases, the state space is still extremely large, and the array of possible actions is vast.

3: COMPUTATIONAL ECOSYSTEM

Sometimes great ideas are birthed before their time, and the concept of a “computational ecosystem” might well be one of these. Some of the earliest work in this area is described by Huberman (1988) in a wonderful little tome called “The Ecology of Computation” containing one of his manuscripts of the same name. The idea germinated at Xerox PARC and presented the idea that multiple agents could be combined into a computational simulation involving multiple, diverse agents working in a complex interaction of processes much like the creatures that inhabit a natural ecosystem. The book also contains a number of other very interesting papers including Miller and Drexler (1988), which develops the idea of agoric or market-driven approach that is central to the technical discussion Section 4. The agoric is the principle that drives the system to stability. For a natural ecosystem the agoric is “survival of the fittest”; for a market economy the agoric is “profit”, and for the StarCraft II development laboratory the agoric is “strategic board position”.

It is important to note that two things are critical for stability in a computational ecosystem: 1) diversity of the actors in the system, and 2) a healthy mix of both cooperation and collaboration comprising autonomous collaboration. This where Hardin’s (1968) argument might be questioned in that perhaps he should have argued that a technical solution that lacks diversity (he had only one type of actor) cannot achieve stability. A computational ecosystem is an example of a complex system in the theoretical sense in that it manifests:

- emergent behavior,
- self-organizing criticality, and

- attractors, representing quasi-stable system configurations

The creatures that live in a computational ecosystem are small programs or “agents” that emulate the maximum extent possible the behavior of their counterparts in the real world. As with a natural ecosystem, each of these cyber creatures maintains their own localized state space which is unique and is used to generate appropriate actions and are organized into an interactive communication network that emulates the web of life. The localized state space is incomplete and contains errors, and the actions often lead to unexpected state changes, hence the complex adaptive system thus modeled is best described as a POMDP. An actor-critic (Ferrari and Stengel, 2004) application of stochastic approximate dynamic programming (ADP) as described in Powell (2011) is used to project from the current local state to a new local state for each unique cyber creature. As in nature, the individual agents have very diverse time horizons that are woven together with a mix of cooperation and competition to make up a stable computational ecosystem.

4: xGRAPH: AN EXECUTABLE GRAPH FRAMEWORK

The foundation upon which a computational ecosystem simulation is constructed is an executable graph framework called xGraph. While an executable graph framework is a much more general tool, it was specifically designed to serve this purpose. It is somewhat unique, sharing some features with a graph database, a distributed programming environment, and an operating system for distributed processing. In this presentation, only those features needed to construct a computational ecosystem are discussed.

An xGraph platform is similar to a modern graph database, such as MongoDB™ or Couch™ in that it comprises a network of nodes that contain data and information, but it is not a data store in the traditional sense in that it does not require external applications to manipulate the information content of the nodes or the relationship among these elements. Instead, an xGraph application is both introspective as well as reflective. It is introspective in that the nodes are capable of sensing the environment as well as approximating the value of being in a particular state, and it is reflective in the sense that it modifies its own data and structures in response to that information. That is, it is capable of being aware of its performance and self-modifying to improve its performance, adapting to changes in the environment as well as the performance or failure of its own components. An xGraph instantiation as described here is an example of a homoiconic system as developed by Kay (1969) in that the components are meant to be high-fidelity analogs of their counterparts in the real world. It is also an autonomic system in the sense described by Kephart and Chess (2003) in that it is 1) self-configuring, 2) self-healing, 3) self-optimizing, and 4) self-protecting. It also adheres to later conditions on autonomic computing as outlined in the Wikipedia (2018) article on Autonomic Computing including such things as learning, self-documenting, and resource management.

5: CURRENT APPLICATIONS

To demonstrate that a technology has practical applications, it is necessary to describe how that technology is actually being deployed in the real world. This section briefly describes several such applications that are currently actively under development or already deployed that address this issue.

Medical Outcome Optimization

The xGraph framework is currently being used by an emerging (currently stealth) hospital support company to solve a problem of finding optimal strategies for linking care with outcomes for a large population of hospitals and clinics. Their product has not yet been announced, so it cannot be discussed in detail, but xGraph was viewed as a perfect solution because it allowed for the sharing of knowledge across a distributed network of medical facilities while at the same time adhering to all requirements of HIPAA and patient-privacy restrictions.

Global Earthquake Seismology

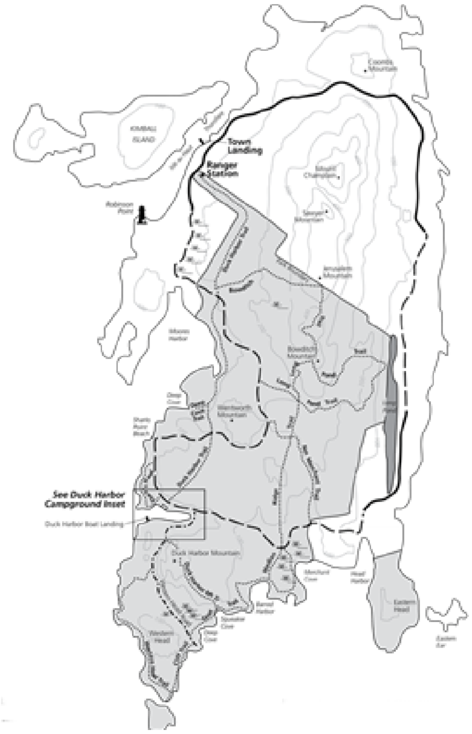
An xGraph solution was developed under contract to the U.S. Geological Survey’s National Earthquake Information Center (NEIC) in Golden, CO to detect and locate all worldwide earthquakes of interest in real time. Here the computational ecosystem comprises millions of hypothesis generators distributed globally that act as a collaborative swarm in analyzing earthquake acoustic data in real time to associate data from seismic events as they occur. This system is currently operational as described by Benz et al. (2015) and virtually every earthquake that is publicly announced was detected and located by the xGraph computational ecosystem running in real-time at the NEIC facility. Extensions of this system are currently being developed under contract to the US Air Force for use in the U.S. and United Nations monitoring of clandestine nuclear testing.

U.S. National Electrical Grid

There is a critical problem with the architecture of the aging US Electrical Grid both from an operational perspective involving the integration of distributed energy resources (e.g. wind and solar) as well as presenting an enormous threat surface for cyber-attack. Currently a localized and fractal approach to modernize the national electrical infrastructure is being developed under contract to the US Department of Energy, Office of Electricity Delivery and Energy Reliability, using xGraph combined with blockchain and cyber currency technologies. The system eliminates the cyber security threat while creating a self-balancing computational ecosystem of electrical loads and generation. A second application of this technology is being used to develop a microgrid control system as an ecosystem of collaborating loads and generation for an island community off the coast of Maine, described in the next section.

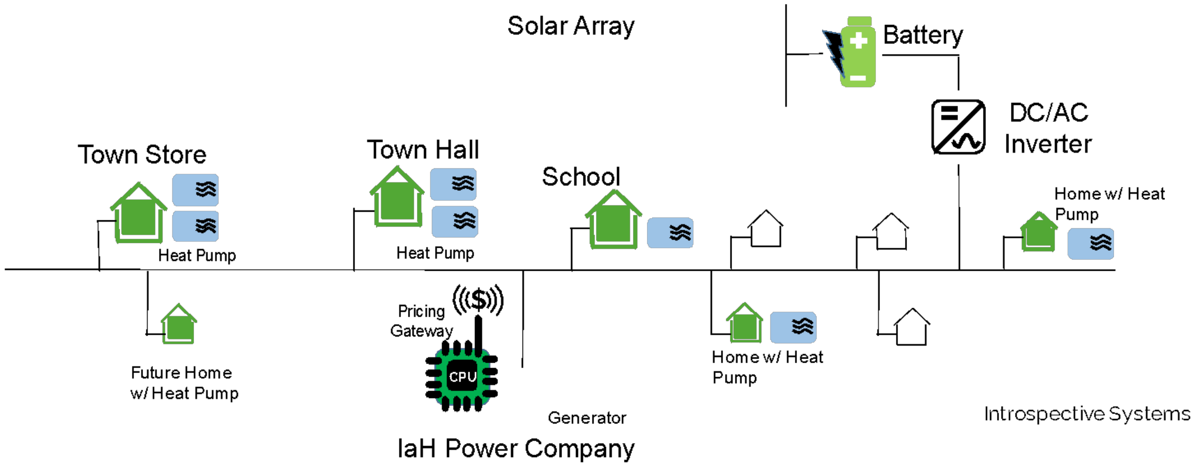
Microgrid Application: Isle au Haut, Maine

Power for Isle au Haut (IaH), Maine is supplied by an aging seven-mile undersea cable. Anticipating cable failure, IaH Electric Power Co assessed many alternatives and concluded that a transition to near-total reliance on solar is by far its best option. Peak electricity demand occurs in the summer when the island’s population is largest. A solar project designed to match this seasonal pattern will generate excess power in the winter. This project installs active load-demand management and air-to-water heat pumps/thermal storage to use excess solar production at optimal times to make a microgrid that meets 100% of the islands needs feasible.



A traditional electrical grid is an on-demand system where generation output must follow load. Loads are uncontrolled; as demand rises, generation needs to keep up. As distributed energy resources (DER, primarily intermittent solar and wind) increase, the predictability of this generation is reduced, making load following even more difficult. To help address this, utility companies are implementing more aggressive demand-side management (DSM) strategies to curtail load. However, these strategies often fall short when variable, intermittent renewable generation is a high percentage of the overall supply. Simple time-of-use (TOU) strategies for commercial customers do not meet the needs of this new dynamic grid with DERs.

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, this is something simpler than others because the behavior of the devices is for the most part predictable (barring failures). 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.



StarCraft II: A Laboratory for Collaborative AI

Late in 2017, something amazing happened: Google™, working with Blizzard Entertainment™, released an API (Application Programming Interface) that allowed bots (a contraction of "robots") to play against other bots or human adversaries in the video game StarCraft II (Knight, 2016; TechCrunch, 2017). This could not have happened at a better time for the technology discussed here. However, as it was released it was purposed as a foundation for developing Google's Deep Learning technology, a centralized approach to complex systems management, and not well suited for developing AI algorithms targeted at collaborative swarms of autonomous entities. At Introspective Systems, we resolved this issue by creating a translation layer for the API that converted it into a computational ecosystem that was well-suited for autonomous swarm AI research. In our implementation, each unit of the StarCraft team is represented as an autonomous system, and makes its own decisions about actions. Thus, we are using the modified StarCraft II environment as a test bed for exploration of dynamic complex systems.

6. SUMMARY

Dynamical complex systems, in which there are many interacting heterogeneous components, can be modeled with the xGraph framework. More powerfully, the xGraph framework and the perspective of dynamical complex systems can be used to implement powerful solutions to the challenges of making effective decisions in complex environments where interacting systems and relevant information evolve even as those decisions are being made.

REFERENCES

Benz, H., C.E. Johnson, J. Patton, and P. Earle (2015) Glass 2.0, **An Operational, Multimodal, Bayesian Earthquake Data Association Engine**, 2015 Fall Annual Meeting of the American Geophysical Union.

Ferrari, S. and R. F. Stengel (2004). Model-Based Adaptive Critic Designs in Learning and Approximate Dynamic Programming, J. Si, A. Barto, W. Powell, Eds., John Wiley and Sons.

Hardin, G. (1968). **The Tragedy of the Commons**, *Science*, No 162. 1243-1248.

Huberman, B. A. (1988). **The Ecology of Computation**. in The Ecology of Computation, ed: B.A. Huberman, North-Holland.

Kay, A. (1969) **A Reactive Engine**, PhD Thesis, University of Hamburg, Hamburg, Germany (retrieved: http://www.mprove.de/diplom/gui/kay69.html, June 14, 2015).

Kephart, J.O, and D.M. Chess (2003), **The Vision of Autonomic Computing**, Computer, 36: 41–50, 2003.

Knight, W. (2016). **StarCraft Will Become the Next Big Playground for AI.** MIT Technology Review. Nov. 4, 2016.

Miller, M. S. and K. E. Drexler. (1988). **Market and Computation: Agoric Open Systems** in The Ecology of Computation, ed: B.A. Huberman, North-Holland.

Powell, W. B. (2011). **Approximate Dynamic Programming: Solving the Curse of Dimensionality, 2nd Edition**. Wiley and Sons.

Senge, P. (2010). **The Fifth Discipline: The Art & Practice of The Learning Organization**, Revised and Updated Edition. Random House.

TechCrunch (2017) **Blizzard and DeepMind turn StarCraft II into an AI research lab**. Downloaded July 19, 2018 from https://techcrunch.com/2017/08/09/blizzard-and-deepmind-turn-starcraft-ii-into-an-ai-research-lab/

Vincent, J. F. V., O. Bogatyreva, N. R. Bogatyrev, A. Bowyer, and A. Pahl (2006). **Biomimetics: Its Practice and Theory**. In Journal of the Royal Society Interface, pp 471–482, 2006.

Wikipedia (2018). **Autonomic Computing**. Downloaded July 18, 2018 from https://en.wikipedia.org/wiki/Autonomic_computing.