Emergence in Stigmergic and Complex Adaptive Systems: A Formal Discrete Event Systems Perspective

Saurabh Mittal, Dunip Technologies, Tempe, AZ, USA

Abstract

Complex systems have been studied by researchers from every discipline: biology, chemistry, physics, sociology, mathematics and economics and more. Depending upon the discipline, complex systems theory has accrued many flavors. We are after a formal representation, a model that can predict the outcome of a complex adaptive system (CAS). In this article, we look at the nature of complexity, then provide a perspective based on Discrete Event Systems (DEVS) theory. We pin down many of the shared features between CAS and artificial systems. We begin with an overview of network science showing how adaptive behavior in these scale-free networks can lead to emergence through stigmergy in CAS. We also address how both self-organization and emergence interplay in a CAS. We then build a case for the view that stigmergic systems are a special case of CAS. We then discuss DEVS Levels of systems specifications and present the dynamic structure extensions of DEVS formalism that lends itself to a study of CAS and in turn, stigmergy. Finally, we address the shortcomings and the limitation of current DEVS extensions and propose the required augmentation to model stigmergy and CAS.

Keywords: Stigmergy; Complex adaptive systems; Emergence; Self-organization; DEVS; Dynamic structure; scale-free networks; Artificial systems

1 Introduction

A natural system is not a monolithic system but a heterogeneous system made up of disparity and dissimilarity, devoid of any larger goal. The system just "is." Examples of such systems include ant colonies, the biosphere, the brain, the immune system, the biological cell, businesses, communities, social systems, stock markets etc. Such systems are adaptable systems where emergence and self-organization are factors that aid evolution. These systems are classified as complex adaptive systems. According to Holland (2006, 1): "CAS are systems that have a large number of components, often called agents that interact and adapt or learn."

In this article, we investigate CAS by looking at the scale of components, interactions between the components, and emergent properties that are manifested by such CAS. We will attempt to understand some of the common underlying properties, address the adaptive nature of such complex systems and illustrate how resilience is an inherent property of CAS.

CAS is occasionally modeled by means of agent-based models and complex network-based models. Multi-agent systems (MAS) is the area of research that deals with such study. However, CAS is fundamentally different from MAS in portraying features like self-similarity (scale-free), complexity, emergence and self-organization that are at a level above the interacting agents. A CAS is a complex, scale-free collectivity of interacting adaptive agents, characterized by high degree of adaptive capacity, giving them resilience in the face of perturbation. Indeed, designing

Email address: saurabh.mittal@duniptechnologies.com [S. Mittal]
Submitted: May 7th, 2012
Revised: May 30th, 2012 (accepted)
Special Issue on Stigmergy, Journal of Cognitive Science Research

an artificial CAS requires formal attention to these specific features. We will address these features and the formalisms needed to model CAS.

The discipline of modeling originated to understand natural phenomena. By developing abstractions, we can manage the apparent complexity, reuse it and enable these complex phenomena in artificial systems to our advantage. The discipline of executing this model on a time base is "simulation." The task of decoding the original structure from manifested behavior is the holy grail of the modeling and simulation (M & S) enterprise (Zeigler, Praehofer, & Kim, 2000). The need for M & S to make progress in understanding CAS has been well acknowledged by Holland (1992). The task is to understand the gamut of rules that exist within and without a component and understand how the component deals with such multidimensional rules in an interactive environment. M & S is the only way one can understand, mimic and recreate a natural system. Most artificially modeled systems that exhibit complex adaptive behavior are driven by multi-resolution *bindings and interconnectivity* at every level of system behavior. To understand life is to "model"; to adapt is to survive in an environment, where both survival and environment are loaded concepts based on the guiding discipline.

Complexity is a phenomenon that is multivariable and multi-dimensional in a space-time continuum. Therefore, what we need is a framework that helps develop system structure and behavior in an abstract manner and that is component oriented so that the system can define its interactions based on the composition of a multi-level environment.

Stigmergy, the study of indirect interaction between network components in a persistent environment, explains certain emergent properties of a system. The network components include both the environment and the agent and both are persistent, i.e. both are situated in a space-time continuum and have memory. We take Stigmergic systems to be a subset of CAS and argue that stigmergic behavior is an emergent phenomenon too. Ultimately, we are trying to get a handle on how to formalize the property of "emergence."

Discrete event abstraction has been studied at length by Bernard Zeigler throughout his illustrious career and his pioneering work on Discrete Event Systems (DEVS) formalism in 1970s (Ziegler, 1976). As a student, his perspectives on CAS were influenced by Holland. Ziegler's approach to CAS has been through the quantization of continuous phenomena and how quantization leads to abstraction. Any CAS must operate within the constraints imposed by space, time, and resources on its information processing (Pinker, 1997). Evidence from neuronal models and neuron processing architectures and from fast and frugal heuristics, provide further support to the centrality of discrete event abstraction in modeling CAS when the constraints of space, time and energy are taken into account. Zeigler stated that discrete event models are the right abstraction for capturing CAS structure and behavior (Zeigler, 2004). In this article, we take the discipline of modeling CAS forward, by looking at the emergence aspect of CAS. We introduce DEVS and demonstrate how recent extensions still fall a little short in modeling CAS.

We first focus on the study of network science and how scale-free networks are inherently important to study complex interactions and hierarchical systems. In Section 3 we look at various types of interactions in a complex network. Section 4 we address the concepts of emergence and self-organization in detail and examine how a complex dynamic network facilitates such behavior. Section 5, a slight digression, provides an overview of DEVS theory. We return to the subject of dynamism in a complex adaptive network in Section 6 and show how DEVS theory is positioned to give modeling and simulation support to the subject. We describe various existing formal DEVS extensions that help model various features of stigmergy, emergence and CAS. Finally, in Section 7, we present some conclusions and pointers for future research.

2 The Nature of Complex Networks

2.1 Overview

Complex networks are the backbone of complex systems and each complex system is a network of interactions among numerous network elements. Some networks are geometric or regular in 2D or 3D space and some have "long range" connections that are not spatial at all. Network topology or anatomy is important to characterize because structure affects function and vice-versa. The dynamic nature of a network is one of the keys to understand complexity. Each network comes with peculiar set of properties and the manifested behavior of the components is bounded by the constraints the network imposes on them (Barabási, 2003). Each network originates through a set of constraints between the nodes that govern how the links are formed. Such constraints, defined as rules, have totally different manifestations when we talk of social networks built by mutual friends, where the rules are dynamic and of a biological cell, where the DNA blueprint along with unchangeable laws of chemistry and physics govern and dictate all the reactions the cell participates in.

Table 1 classifies networks in complex systems along with three metrics that are used to compare them. The first metric, *average path length (L)* is used to measure the smallest number of edges connecting nodes A and B. In a fully connected graph, L equals 1. The maximal path length is called the *network diameter* (D). The *degree* (K) of a node A is the number of its connections or nearest neighbors. The second metric, the *degree distribution P(k)* is the probability distribution of the node degrees and shows their spread around the *average degree*. A *neighborhood* is the set of K nodes at distance 1 from node A. The third metric, the *clustering coefficient* (C) (Watts & Strogatz, 1998) is the ratio of node A's neighborhood with all possible connections from A. The maximum value C is 1.

Network Type	**Description**	**Examples**	**Structural Metrics (for nodes N)**		
			Average Path Length (L) and Diameter (D)	**Degree Distribution (connectivity) *P(k)***	**Clustering coefficient (C)**
Regular 1. Fully connected, 2. 2D or 3D-lattice, 3. ring-world lattice 4. constaints-based	Each node is connected to other node based on a specific type of sub-property such as geometric constraint or any defined constraint. Fully connected has the highest number of edges.	Slime mold, animal coats, insect colonies, bird flocking, swarm sync	Fully connected: Lowest L=D=1, 2D,3D- lattice and ring world: D>>1, L ~ N when K<<N	Fully connected: K = N-1, P(k) = f(k-N+1) 2D,3D-lattice and ring world: Low, P(k) = f(k-K)	Fully connected: Highest, C = 1 2D,3D-lattice and ring world: High, C ~ 0.75

Small-world	Same family of regular networks with few links breaking the 2D or 3D or ring-world lattice such that there is no loss in clustering, however reducing path length on a log-scale	Hollywood, web pages, social	$L \sim \ln N$	Poisson degree distribution	High, $C \sim 0.75$ for $K >> 1$
Scale-free	Built on small-world networks with high vulnerability to targeted attacks. Their origin can be attributed to two properties: growth and preferential attachment	Ecology, Internet, brain, biological cell, gene regulation, airline, citation, metabolic, power grid, language, economy	$L \sim \ln N$	Power law[1] (Newman, 2005)	High, $C \sim 0.75$ for $K >> 1$

Table 1: Classification of complex networks and their metrics

We put more focus on the scale-free networks as they are the most complex of the three types. Such complexity is evident in various systems that exhibit scale-free behavior (shown in Table 1, column 3) (Barabási, 2003). These systems evolve towards scale-free topology to become resilient and are sometimes classified as CAS due to the resulting behavior.

2.2 Scale free Networks

In most complex networks found in nature, the nodes are dynamic agents that extend themselves in an environment to build links with either the environment's objects or other agents. These agents are dynamic in the sense that they govern their interactions with their neighbors through a dynamic unpredictable environment. As nodes are added incrementally, a very important phenomenon occurs when the network reaches a critical state. It undergoes a state change wherein it becomes subjected to a new set of rules, i.e. it starts displaying a totally different behavior that it did not manifest initially. These fundamental shifts are known as "giant component" in mathematics (Bradonjic, Hagberg, & Percus, 2007), percolation or phase transition in physics, or simply a community in sociology (Barabási, 2003). Critical state transitions impact the behavior of agents, which in turn impact the link structure and consequently, the network topology, bringing in new affordances and constraints altogether.

In order to understand why critical state transition occurs, we ask the following:

I. How are the rules inside these agents triggered when "critical state transition" occurs?
II. Does the agent already have an inherent capacity and does the presence of a conducive environment enforce the agent to manifest a totally different behavior?
III. Does the agent learn new rules by active observation in an environment?
IV. Do inherent capabilities make the agent vulnerable to external influence?

[1] Power law: a slow continuously decreasing curve implying that many small events coexist with few large events. Each power law has a unique degree exponent. $P(k) \sim K^{-Y}$, where Y is the degree. In most of the networks, the number of nodes with exactly K links follows a power law, each with a unique degree exponent Y between 2 and 3.

These questions raise an understated assumption. That is, agents are a part of an interconnected network, the environment is a part of the network and the interactions exist between the agents and the environment. Defining boundaries is therefore rather problematic because weak links and non-linear interactions affect other agents and the environment in unpredictable ways. These are open systems, where agents are connected both directly and indirectly: CAS are open systems.

A group of highly connected nodes is called a *cluster*. A network is composed of these clusters and weak ties connecting these clusters. Weak ties play a crucial role in network formation. It is only through these links that new information is acquired by the original network as the strong ties stay within the cluster (Barabási, 2003; Watts, 1999). Society is made of many highly connected clusters connected together by weak ties. This came to be known as the *small-world* effect. Newman's work (Newman & Watts, 1999) along with Barabási offered quantitative evidence that clustering is present in social systems and is ubiquitous in nature, thereby making it a unique organic property of complex networks. Clustering coefficients measure the density of a particular cluster: it was found that a few links were sufficient to reduce the average separation of the nodes without making an impact on the clustering coefficient keeping it practically unchanged (Watts & Strogatz, 1998). The "Six Degrees of Separation" experiment (Milgram, 1967) is famous for describing such phenomenon in social networks. In a cluster, some nodes may act as a *connector* or a *hub*, i.e. nodes with an anomalously large number of links. Such hubs are present in every complex system, whether it be a financial system or a cellular network and is another fundamental property of complex networks. The formation of hubs and clustering in a dynamic network redefines the dynamic landscape of a complex system. In society, these are people with varied interests, who connect myriad fields, expertise and experience.

Power law degree distributions are scale-free and most natural complex systems have power-law behavior (Buchanan, 2001; Newman, 2005). Each scale-free network has hubs that fundamentally define the network's dynamic topology. In mathematical terms, power law is a notion that a few large events carry most of the action. So, another question is:

V. What makes a node or an agent evolve into a hub?

In order to answer this question, scientists looked at physics of atoms e.g. the emergence of magnet (Stanley, 1971), is a phase transition from *disorder* to *order*. Wilson's theory of "renormalization" (Wilson, 1971; Kadanoff, 1993) started with scale invariant behavior and assumed that at the critical point, the laws of physics become applicable in identical manner across all scales, from atoms to boxes containing millions of identical atoms, all acting in unison. He proved mathematically that at such an instant, power laws emerged bringing forth order from chaos. Power laws, as another organic property of complex networks, are patent signatures of self-organization in complex systems.

All nodes are not equivalent. As nodes with their preferences and biases acquire links, their behavior seems to facilitate more link making, i.e. they start portraying affinity for new links. Eventually, they become hubs. Real networks are governed by two laws: growth and preferential attachment, i.e. the network is dynamic, links/nodes are dynamic (they can appear/disappear/rewire) and each link has a probability towards a high affinity node (Newman, 2001). This preferential attachment is a rule that governs how a network is formed in the first place, from individual nodes leading to a scale-free topology (Jeong, Neda, & Barabási, 2003). Most complex networks of scientific and practical importance are scale-free, for example, metabolic network within cell (Jeong, et al., 2000), citation networks (Bilke & Peterson, 2001), economic webs, language networks and many others.

In most complex networks, each node has unique properties and behaviors that are apparent even if its connectivity is unknown. Interestingly, it is these intrinsic properties that partially define and decide what connectivity this node will eventually have in a competitive environment. This competition (may be linked to survival) in a heterogeneous system, decides winners and losers. The rate at which any node acquires connectivity, gaining edges to become a hub defines the new topology and its impact factor in the current network. This rate is the quantitative measure of node's ability to stay ahead of competition, called "fitness." Preferential attachment is driven by the product of the node's fitness and the number of links it has. This fitness function allows a late-coming node (e.g. Google) in the network to impact old connectivity and reorganize the entire topology towards itself, making it a hub. Theoretically, a hub may form that can grab all the links in the existing network, such as in the phenomenon "winner take all," and totally redefine the landscape to an extent that other node's behaviors do not matter. It destroys other hubs and makes a network star-like, i.e. the property of the complex network totally gets transformed (Barabási, 2003). In an interconnected world, links represent survival and competition exists between nodes. In a stigmergic system, interaction is the very mechanism that defines this interconnectedness.

Most real-world systems are generally resilient and their functionality is guaranteed by a highly interconnected complex network. Resilience in a scale free network is rooted in its topology (Barabási, 2003). A significant fraction of nodes can be randomly removed without breaking the network apart when these nodes are not the hubs. Any targeted attack to these hubs can disintegrate the network and reduce it to independent clusters. Such inverse phase transition is evident in cascading network failures, distributed denial of service (DDoS) attacks, grid failures, avalanches, etc. where the load handled by a node is transferred to other neighboring nodes that are not prepared to take such load. Such a phenomenon is also known as *self-organized criticality*. Not understanding how the actions of one node affect other nodes can inadvertently make the network vulnerable by hitting the hub, leading to cascading failures and complete breakdown of the topology. In a globalized economy the strength of such links becomes more relevant as the supplier and the buyer are not competitors but partners sharing the burden of their respective networks. The financial crisis (IMF, 2009) that began in 2007 is evidence how the failure of one large bank (Treasury, 2007) risked an entire global financial system. Several major banking institutions either failed, were acquired under duress, or were subject to government takeover. These included Lehman Brothers, Merrill Lynch, Fannie Mae, Freddie Mac, Washington Mutual, Wachovia, Citigroup and AIG (Altman, 2009).

Diffusion and spreading in a complex network is described by a threshold model. Each node has a critical threshold that allows or prevents it from communicating the idea or message to the next neighbor. Intuitively, if a hub has a lower threshold, such messages are communicated far; the message is not censored and filtered. On the other hand, other nodes that have a higher threshold will prevent the spreading of the message. This distribution of critical threshold levels within the network nodes is a property of the network that explains phenomena such as virus spread, fire spread, adoption of innovative ideas and so forth.

Multi-tasking and concurrency is another inherent property of most complex systems, i.e. each entity continues to display its behavior in the network environment and affects it. Each network node whether simple or complex, is modular. Modularity is an essential property and a defining feature of a complex network wherein it is defined through a node's interface, what it brings to a network in qualitative terms. Each node brings value to the network to keep the network functional. Real networks are clearly scale-free and modular at the same time. Quantitatively, clustering coefficient measures modularity. However, as clusters are formed and hubs are born, the coefficient takes on an inverse behavior, i.e., as hubs emerge they reduce their number of links. Their role transforms from being a functional participant to a more structural role where they maintain links with other hubs to enable connectivity with other clusters, giving rise to weak links across a scale-free network (Barabási, 2003). A hierarchy is born and it allows the evolution of these modules independently. Acquiring a status of hub entails change in its dynamic interface that now affords new links and messages to come across.

Real-world networks are self-organized where independent actions of the constituent nodes and links lead to emergent behavior. The robustness of the laws governing the emergence of complex networks are not confined to a special class of systems but are rooted in the properties and behavior of the nodes guided by two basic conditions of *incremental growth* and *preferential attachment*, necessary for a scale-free topology. Adding two more questions to the list:

VI. What dynamics happen within a node that urges it to extend a link, and
VII. What transpires through when a link is formed between two nodes?

This section raised some questions on the very foundation of a scale-free complex interactive system. How to determine the source of dynamism and how dynamism affects other nodes and shape the network landscape towards a self-reliant complex network, are issues that are further investigated ahead.

3 The Nature of Actions in a Complex Adaptive System

In this section we take a look at different types of actions that impact the evolution of the network. We classify these actions into two broad categories, i.e., intra-actions and inter-actions.

Intra-actions: These are the actions taken by the node internally, i.e., these actions impact the node itself first and may impact other nodes through various interactions this node participates in. These actions are initiated by the internal dynamics of the node.

Inter-actions: These are the actions taken by the node in an external environment, i.e., these actions impact other nodes in the network. This impact to other nodes is communicated through various modes of communication as allowed by the environment or specified by the properties of the network. Such interactions have been further classified by Keil & Goldin (2003) as:

a. *Direct interactions*: These interactions are point-to-point connections and the actions of one node directly affect other node and *nobody* else.
b. *Indirect Interactions*: These interactions are publish-subscribe phenomenon where a node publishes its actions (through specific messages) to a shared environment (acting as a

persistent medium) and other nodes that are subscribed to this particular type of messages, becoming affected through the medium of exchange.

In the real world, each of agent nodes and the environment is persistent, i.e. they have memory. In modeling complex dynamic systems, we must take memory of both the agents and the environment into account. If agents simply respond to a persistent environment and have a fixed set of rules devoid of any complex learning and/or memory apparatus, then agents are reactive. Stigmergic behavior can still occur in such an environment. However, for an agent to transform from a node to a hub, advanced learning and memory apparatus must be present and that can be triggered when critical state transition occurs.

Klein and Goldin (2003) prove that the behavior of computational agents that make use of indirect interactions via the real world are richer than the behavior of agents that interact directly. Their proof is based on the result by Siegelmann (1999): that the real world that is analog in nature may compute algorithmically incomputable functions. Persistence plays a crucial role in the definition of indirect interactions. Persistent Turing Machines (PTM), equivalent to the Interactive Transition Systems (ITS) model combines the constraints of computable transitions, as in Turing machines, with the extension to non-algorithmic, interactive processing associated with persistent state (Golden, et. al, 2001; Golding, 2000).

3.1 Stigmergy as a Complex Adaptive System

Stigmergy is a concept that describes how self-organization of nodes emerges in a persistent medium through an indirect communication. Self-organization is the interaction of a set of processes at a lower level of system to yield structures at a higher global level (Holland & Melhuish, 1999). This does not imply that self-organization evolve into a hierarchical organization. For a hierarchy, i.e. a hub to appear, the self-organization must be coupled with other constructs such as scale-free nature of underlying network, coherent emergents, and upward as well as downward causations. This demarcates the boundary between a stigmergic system and a CAS. Self-organization leading to hierarchy has been observed in most natural CAS.

Indirect interaction is crucial to any stigmergic system to occur in the first place. This kind of behavior was first observed in ant colonies by Pierre-Paul Grassé (1959) who was intrigued to learn how these virtually brainless creatures could create highly sophisticated messaging systems and build extremely complex architectural structures. According to Grassé, complexity in a stigmergic system arises because the individuals interact not with each other but through an environment by making changes to the environment that have spatiotemporal character to them. These spatiotemporal functions in the environment impact other individuals and the causal node itself situated in the same environment. The creation of positive and negative feedback loops, amplification of fluctuations in the presence of multiple interactions shape both the individual and the environment (Camazine, et al., 2001). An individual makes local changes in its environment that last long enough for other individuals to detect and be affected by them. Such behavior has also been researched in robotic MAS that display *swarm behavior*. One of the features of MAS and the contained autonomous agents within that exhibit indirect communication is the notion of "decentralized" control (Doyle & Kalish, 2004). Assumptions that all well-performing systems have "leaders" and centralized control are proven erroneous (Resnick, 1994; Sulis, 1997). Collective robotics is another case where stigmergic concepts are applied (Izquierdo-Torres, 2004). Similar work by Brooks (1991) acknowledges the fact that the fundamental decomposition of intelligent systems is not in the identification of individual processing producers that must interface with each other, but in the interactions that these producers have with the world through perception and action.

The capacity to detect and be affected by changes in the environment as a result of an action of another node in a spatiotemporal manner is guided and simultaneously constrained by both the properties of a network and the internal structure of the node. The capacity "to detect" is a property analogous to preferential attachment and affinity. The capacity "to be affected" is a property analogous to the threshold model. The third property is "to form weak links" with other individuals separated by space and time. Such weak links in a stigmergic system are made possible through the persistent environment and the persistent initial state of the agents. These three properties are also central to any CAS. However, there are some CAS properties, such as scale-free nature, and the underlying network complexity that may or may not be portrayed by a stigmergic system. This makes a stigmergic system a special case of a CAS:

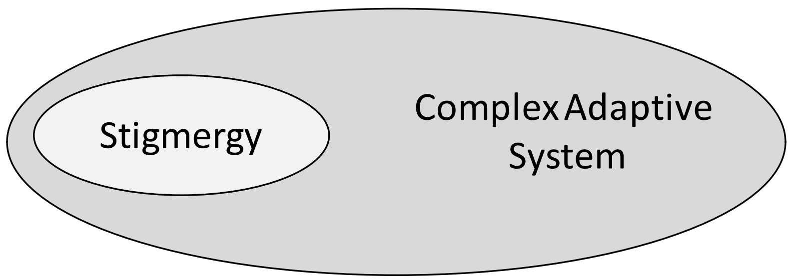

Figure 1: Stigmergy as a special case of CAS

4 Emergence and Self-organization

Emergence, a term coined by Lewes (1875) has gained widespread attention in the last two decades partly due to the analysis capabilities afforded by massive computational power and partly due to widespread complex systems in everyday use such as World Wide Web. Emergence has been a native of the land of "complex systems" and there are four schools of thought that study emergence, as summarized by Wolf & Holvoet (2005):

1. *Complex Adaptive Systems theory*: Concept of macro-level patterns arising from interacting agents
2. *Nonlinear Dynamical Systems theory and Chaos theory*: Concept of attractors that guide the system behavior
3. *The synergistic school:* Concept of order parameter that influences which macro-level phenomena a system exhibits
4. *Far-from-equilibrium thermodynamics:* Concept of dissipative structures and dynamical systems arising from far-from-equilibrium conditions.

We focus mostly on the emergence as addressed in the complex adaptive systems literature.

4.1 Strong and Weak Emergence

We begin with a definition of emergence as given in (Wolf & Holvoet, 2005, 3):

> A system exhibits emergence when there are coherent emergent at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel with respect to the individual parts of the system.

Other definitions of emergence can be found in (Deguet et al., 2006). Banabeau & Dessalles (1997) in their definition give significant importance to the detection of "something." They also described how emergence is handled in a hierarchical complex system:

> When a detector becomes active in such a hierarchy, the active detectors from the lower level that are connected to it can be omitted from the description. […] Emergence is thus a characteristic feature of detection hierarchies (Banabeau & Dessalles, 1997, 5).

We acknowledge that *emergence* is an observer phenomenon and would add that the *emergence* happens only at levels above the interacting agents, i.e. a hierarchy must be present to detect emergence. The observer is always at a higher level of perception to detect something emerging. Revisiting the ant colony example, an ant doesn't know that it is building a colony. It is the human observer or a higher animal or a detector agent that witness the existence of such structural functional affordances. While these affordances may be present in the environment that would result in the manifested behavior, an observer must be present to label such an "affordance" in an artificial system that self-organizes at the collective behavior at the level above it. In an artificial system, the notion of *observer* takes a stronghold as various observers can be defined at various levels of a hierarchical system that can detect the occurrence and variations in key indicators at specific levels of resolution to categorize as *emergence*. Another way to look at it is through the identification of such key indicators. These key indicators are not a part of the system they are meant to observe. Sometimes these indicators can be derived from the lower level constructs and sometimes they are beyond such deduction and are completely novel (Muller, 2003). This demarcation in detection capabilities was formalized by Banabeau & Dessalles (1997) as $L_{(n-1)}$ and L_n levels. $L_{(n-1)}$ is a level where individuals interact in a persistent environment. L_n is a level above $L_{(n-1)}$ that observes these global collective properties of $L_{(n-1)}$. Both $L_{(n-1)}$ and L_n have their own grammar and representation. Obviously, if a pattern in a lower level $L_{(n-1)}$ is not formalized as a detector at L_n then such pattern will fail to "emerge" at L_n. This correlates with deducible emergence (Baas, 1994) where two disjoint levels are linked by a computational process i.e. how $L_{(n-1)}$ and L_n interact computationally. Alternatively, we as humans see what we are entrained to see through our rich set of experiences. If we are unable to identify a pattern, it is highly unlikely we will label it as such, i.e. we don't understand the grammar at L_n!

The other viewpoint from Bonabeau and Dessalles (the second quote) is congruent with the nature of complex networks that explains how hubs are formed in the real world. To further analyze this congruence, one has to put the act of observing in the agent itself. When we design intelligent agents, we could also imbue them with the power to observe the collective whole. Such power of observing the collective impact of other agents indirectly is readily present in a stigmergic system where both the environment and the agents are persistent. Their actions are based on this persistence wherein the collectives have been realized in the environment in a spatiotemporal matrix. Alternatively, agents in a stigmergic environment need not be encoded with complex detection patterns. Just the presence of these agents in a stigmergic scale-free complex system fulfills the requirement of an observer within an agent. So, it can be argued that a stigmergic system provides a foundation for emergence to occur. Now, having both a stigmergic system and a sophisticated detector mechanism in an artificial agent, will make this agent more competitive to other agents due to its handling of additional percepts. As the emerging properties in the system continue to be detected, the relative strength of these properties can guide, in real-time, the role of this agent and various contingencies that are enforced in the environment as a result of change in an agent's behavior. From a performer at a specific level of system, on detection of emerging patterns, agents can become, an enabler, a hub. This is the rise of hierarchy in a self-organized manner as has been described in Barabási's work and the coherent property (Wolf & Holvoet, 2005) mentioned in the first definition of emergence. This aspect of having the act of observation inside the agent is also congruent with ideas of Strong and Weak Emergence by Muller (2003). Muller also added that in Strong emergence, the observer has causal powers.

The system displays strong emergence when the emergent behavior is irreducible to either the agent or the environment as both interact in a dynamic spatiotemporal manner. The emergent behavior also has a downward causation at lower levels (Chalmers, 2006) changing the very nature of the network beneath it. This marks the rise of hierarchy and the critical state transition. In the case of weak emergence, the emergence phenomenon is reducible to its constituent components and effect of causality on lower-levels is questionable. Examples like the "Game of Life" and connectionist networks are examples of weak emergence where laws encoded in low-level rules result in high-level structures and patterns. Basic laws result in unexpected behavior that surprise us, that can be reduced to simple rules (albeit, with difficulty but possible) with no downward causation to impact low-level rules. A stigmergic system displays weak emergence when the agents do not develop hierarchy, i.e. do not become a hub and impact the topology. Ant behavior is example of weak emergence: an ant has basic rules but no ant becomes a hub. A stigmergic system displays both *strong* and weak emergence depending upon the role of observer in *strong* emergence. The collective behavior is manifested in the persistent dynamic environment that the agent is part of. The agent detects such changes and acts over them.

4.2 Self-organization and Emergence

We briefly discussed the formation of a hub in a stigmergic system. We now look at the concept of self-organization within the context of emergence, defined by Wolf & Holvoet (2005, 7) as: "Self-organization is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control."

Note the keywords: dynamical, adaptive, acquire, maintain, structure, and without external control. The context can easily be understood with respect to the scale-free network and their emerging topologies. Wolf and Holvoet (2005) acknowledge the identification of "boundary" of the system which relates to the modularity principle in scale-free networks.

While both emergence and self-organization are dynamic properties of complex systems, the network's and the agent's robust internal properties together will decide if both of them are simultaneously portrayed by the system. Three cases arise (Wolf & Holvoet, 2005, 9-11):

1. Self-organization without Emergence
2. Emergence without Self-organization
3. Emergence and Self-organization together

The first case is easily understood by looking at classical multi-agent systems in which the entities self-assemble to do a particular task. Removal of an entity does not impact the performance of the task because of the adaptive nature of such assembly, as another node can replace the lost node: the system can continue to display the same behavior. There is no novelty in the global behavior, hence no emergence. The network may or may not be scale-free. Growth and decay are not properties of self-organization when considered in isolation. Self-organization is an adaptive collective behavior.

The second case, emergence without self-organization, can be understood by an example of a stationary gas, in the realm of physics. A stationary process is time-translation invariant: a gas has a certain volume in space. This *volume* is its emergent property is novel and unique as a function of its atoms. A system can exhibit chaos that emerges from interaction of these atoms but no self-organization as they do not organize to perform a collective behavior.

The third case, the most interesting one as far we are concerned, is found in natural systems where both the persistent environment and producing agents are present. In such a dynamical system, the agents can hierarchically self-organize and display adaptive emergent phenomena through their defined actions that result in dynamic scale-free topologies. Providing structure to such an emergent complex adaptive system *apriori* is almost impossible as the system structure itself is based on persistent nature of components, dynamic interactions and the resulting topology. As has been evident in evolution of scale-free networks that the hubs tend to reduce complexity (Barabási, 2003), similar results by Shalizi (2001, 118) are present in complexity research that state "… self-organization increases statistical complexity, while emergence, generally speaking, reduces it".

Because these systems are intricately linked, they display non-linear behavior (Heyligen, 2002; Camazine, et al., 2001) where a small perturbation can lead to a large effect due to amplification by positive or negative feedback loops. Network complexity can propagate to a large portion of network, triggering cascaded effects. When caught up in a positive feedback loop, the system will realign when it encounters a similar negative feedback loop. Consequently, in a self-organizing emergent system, the interplay of positive, negative feedback loops, amplifications, suppressions, taken all together as preferential attachment, thresholds and affinities, become a function of the environment and the agent taken together. In a complex adaptive self-organizing emergent system, the system continues to redefine the topology, display the emergent properties and refine the properties themselves. In fact, these properties may very well enable such self-organizing and emergent properties in the first place.

5 Discrete Event Systems (DEVS) theory and its Variants

The Theory of Modeling and Simulation was first introduced in (Zeigler, 1976). Some notable extensions of the original DEVS formalism are Fuzzy DEVS, Dynamic Structure DEVS, Confluent DEVS, Symbolic DEVS and Real-time DEVS. DEVS concepts have been applied to almost every natural phenomenon, from simple state machines to non-linear systems to continuous systems to complex hybrid systems (Zeigler et al., 2000). The depth of DEVS systems formalism was acknowledged by researchers like (Vangheluwe, 2000) who established DEVS be the common denominator of all modeling formalisms due to its mathematical foundation and rigor (see Figure 2). For more details on each of the formalisms, see Vangheluwe (2000).

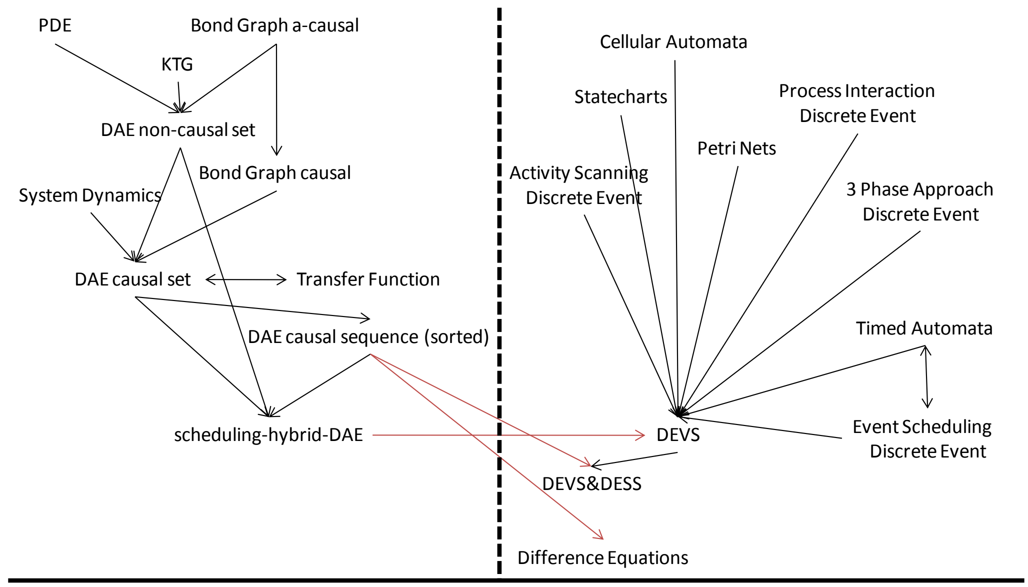

Figure 2: Formalism Transformation Graph

DEVS theory is made up of two orthogonal concepts (Zeigler et al, 2000):

1. *Levels of System Specification:* these describe how systems behave;
2. *System Specification Formalisms:* these incorporate various modeling styles, such as continuous or discrete.

Systems theory distinguishes between system structure (how the system is constituted internally) and system behavior (how the system manifests externally). Understanding the system structure allows us to deduce its behavior. The internal structure of a system is laden with many concepts, such as:

1. *State representation*: different states the system may exist in
2. *Transition functions*: mechanisms that allow moving from one state to another
3. *State to output functions*: mechanisms that make system a producer in an environment
4. *Composition*: capacity to form a larger system by coupling smaller systems
5. *Decomposition*: capacity to decompose into smaller systems from a larger system
6. *Hierarchical construction*: capacity to continue to portray composition
7. *Modular*: capacity to have defined input and output interfaces to enable composition

Systems theory is *closed under composition* in that the structure and behavior of a composition of systems can be expressed in original system theory terms. This is the foundation of modular systems that have defined input and output interfaces through which all interaction with the environment occurs. Such modular systems are coupled together to form larger ones leading to hierarchical construction.

More detailed description of alternate system specification formalisms such Differential Equation System Specification (DESS), Discrete Time System Specification (DTSS), and Quantized Systems can be found in (Zeigler, et. al, 2000). While DESS and DTSS as their names suggest are self explanatory, Quantized DEVS warrants a definition. Quantization is a process for representing and simulating continuous systems as an alternative to the more conventional time axis. It is built on *threshold crossing* model. While *discretization* leads to DTSS, *quantization* leads to discrete event systems. Using quantization to develop abstractions for CAS is also addressed in (Zeigler, 2004). Figure 3 shows a mapping of various formalisms. As can be seen, the universality of DEVS formalism allows specification of hybrid systems.

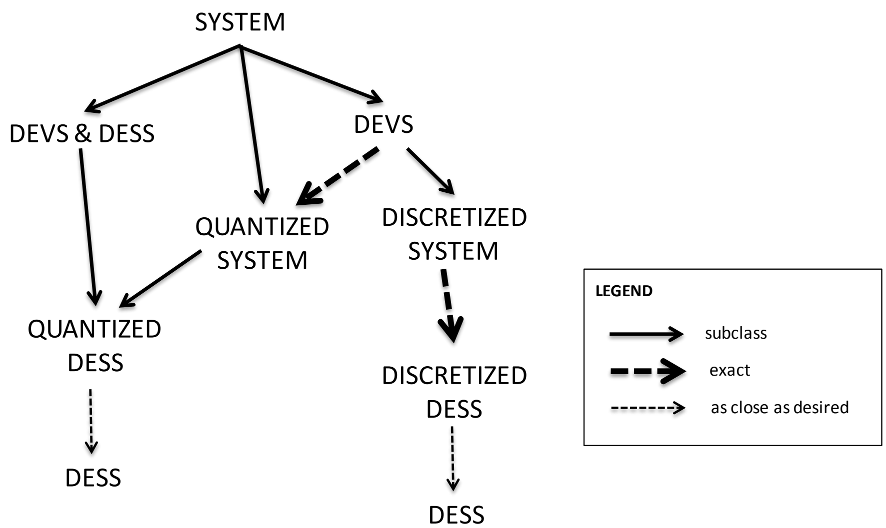

Figure 3: DEVS Formalism and Quantized Systems. Reproduced from Zeigler, et al, 2000.

Having seen the scope of the DEVS formalism, we focus our attention on the DEVS formalism and levels of system specifications.

5.1 Levels of System Specification

The DEVS levels of system specification has 5 levels. At the most basic Level 0 is the Observation frame that defines which inputs stimulate the system, what variables to measure and how to observe them over a time base. At this level we also think about the possible range of values the inputs may take. The observation also correlates the input trajectory with specific outputs the system produces. Such correlation between inputs and outputs is called an *I/O pair* linked over a time-base. A collection of such I/O pairs is called an *I/O behavior* at Level 1. It is entirely possible that two or more input trajectories may lead to the same output trajectory. At Level 1 we have all those trajectories collected and at level 2 we distinguish them based on the initial state of the system when input is injected. The initial state determines a “unique” response to any input and is represented as an *I/O function*. At Level 2 we not only can define the initial state but also state transitions when the system responds to input trajectories. At Level 3 we have a system that has a state-space and characteristic functions that map specific input trajectories to specific output trajectories. The black box system at Level 3 is an “atomic” component in DEVS parlance, capable of dealing with external inputs and that undergoes state transitions to produce external outputs. At Level 4 we have coupled and interactive systems that are connected using “coupling” relationships. The systems are coupled using ports and outputs of one system and are connected to inputs of another system. Such coupling allows composition and hierarchical construction.

Table 2 summarizes these levels.

Level	Name	System Specification at this level
4	Coupled Systems	Systems built from component systems with a coupling recipe
3	I/O System Structure	System with state and transitions to generate the behavior
2	I/O function	Collection of input/output pairs constituting the allowed behavior partitioned according to initial state of the system
1	I/O behavior	Collection of input/output pairs constituting the allowed behavior of the system from external black-box view
0	I/O frame	Input and output variables and port together with values over a time base

Table 2: Levels of Systems specifications (reproduced from Zeigler, et al, 2000).

5.2 DEVS System Components

Structurally hierarchical DEVS system is composed of three elements – atomic, coupled and associated couplings between the atomic and coupled components.

5.2.1 Atomic component

An atomic parallel DEVS (*parallelDEVS*) M (Zeigler, et al, 2000) is specified by a 8-tuple

$M_{pDevs} = < X_M, Y_M, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta >$, where

$$
\begin{array}{ll}
X_M = \{ (p,v) | p \in IPorts, v \in Xp \} & set\ of\ input\ ports\ and\ values \\
Y_M = \{ (p,v) | p \in OPorts, v \in Yp \} & set\ of\ output\ ports\ and\ values \\
S & set\ of\ sequential\ states \\
\delta_{int}: S \rightarrow S & internal\ transition\ function \\
\delta_{ext} : Q \times X_M^b \rightarrow S & external\ transition\ function \\
\quad Q = \{(s,e): s \in S, 0 \le e \le ta(s)\} & state\ set\ including\ elapsed\ time \\
\delta_{con} : S \times X_M^b \rightarrow S & confluent\ transition\ function \\
\lambda : S \rightarrow Y_M^b & output\ function \\
ta : S \rightarrow R_{0,\infty}^+ & time\ advance\ function
\end{array}
$$

There are no restrictions on the sizes of the sets, which typically are product sets, i.e., $S=S_1 \times S_2 \times ... S_n$. In the case of the state set S, this formalizes multiple concurrent parts of a system, while it formalizes multiple input and output ports in the case of sets X and Y. The time base T is not mentioned explicitly and is continuous. For a discrete-event model described by an atomic-DEVS M, the behavior is uniquely determined by the initial total state $(s_0, e_0) \in Q$ and is obtained by means of the following iterative simulation procedure (see Figure 4).

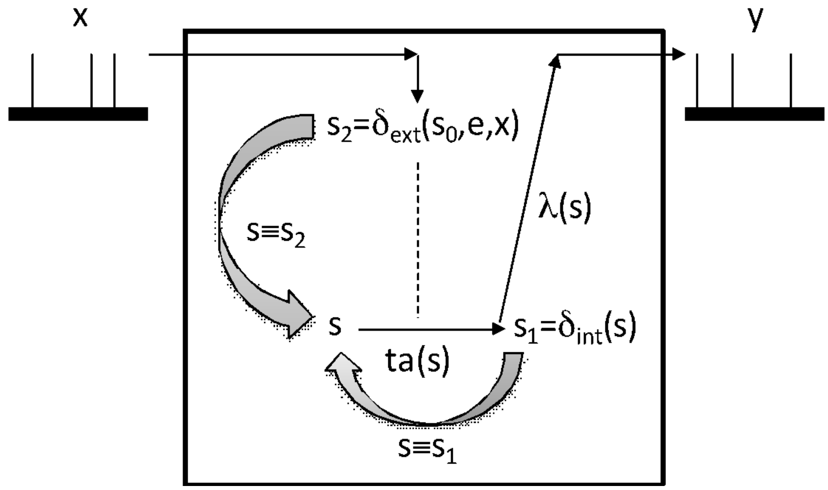

Figure 4: State transitions of an atomic DEVS model.

At any given moment, a DEVS model is in state $\boldsymbol{s} \in \boldsymbol{S}$. In the absence of external events, it remains in that state for a period of time defined by $\boldsymbol{ta(s)}$. When $\boldsymbol{ta(s)}$ expires, the model outputs the value $\boldsymbol{\lambda}$(s) through a port, and it then changes to a new state s1 given by $\boldsymbol{\delta_{int}(s)}$. This transition is called an internal transition and describes the intra-action defined in Section 3. Then, the process starts again (bottom gray arrow). On the contrary, an external transition may occur due to the reception of external events through input ports. This describes the actions taken by the DEVS model when it receives an external input as a result of inter-action (defined in Section 3) by other agents or environment. The external transition function determines the new state s_2 given by $\boldsymbol{\delta_{ext}(s, e, x)}$, where *s* is the current state, *e* is the time elapsed since the last transition (external or internal), and *x* is the external event received. After an external transition, the model is re-scheduled and the process starts again (left gray arrow), setting the elapsed time *e* to 0. In the situation when the internal transition is about to happen and an external input is received, the $\boldsymbol{\delta_{con}(s, e, x)}$ selects either $\boldsymbol{\delta_{int}}$or $\boldsymbol{\delta_{ext}}$. From a structure perspective, an atomic model is made of set of input ports and values specified as $\boldsymbol{X_M}$, a set of output ports and values as $\boldsymbol{Y_M}$, and a non-empty set of states $\boldsymbol{S}$. From a behavior perspective, there are the $\boldsymbol{\delta_{int}, \delta_{ext}, \delta_{con}, and\ \lambda}$.

To summarize: *sigma* holds the time remaining to the next internal transition. This is precisely the time-advance value to be produced by the time-advance function. In the absence of external events the system stays in the current state for the time given by sigma. The time advance function can take any real number between 0 and ∞. A state for which ta(s) = 0 is called transient state. In contrast, if ta(s) = ∞, then s is said to be a passive state, in which the system will remain perpetually unless an external event is received.

5.2.2 Coupled Component

A coupled model (Zeigler et al., 2000) N is described by an I/O interface X, Y, a set of components D, and a set of couplings between those components internally (IC) and the parent coupled model (EIC and EOC). EIC is the external input couplings that couples N with internal component $M_d \in D$, and EOC is external output coupling that couples $M_d \in D$ to N. No self-coupling is allowed i.e. a models output ports cannot connect to its inports. Mathematically, a coupled model N is described by:

$N_{pDevs} =< X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC >$, where

$X = \{(p, v) | p \in IPorts, v \in Xp\}$ is the set of input ports and values,
$Y = \{(p, v) | p \in OPorts, v \in Yp\}$ is the set of output ports and values,
D is the set of component names,
$M_d =< X_d, Y_d, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta >$ is a DEVS,
$EIC \subseteq \{((N, ip_n), (d, ip_d)) | ip_N \in IPorts, d \in D, ip_d \in IPorts\}$ $EOC \subseteq \{((d, op_d), (N, op_N)) | op_N \in OPorts, d \in D, op_d \in OPorts\}$
$IC \subseteq \{((a, op_{a)}, (b, ip_b)) | a, b \in D, op_a \in OPorts_a, ip_b \in IPorts_b\}$

This formal description leads to development of hierarchical DEVS models where subcomponents can be either coupled or atomic with defined interface specifications (Figure 5). M1, M2, and M3 can be either an atomic model or a hierarchical coupled model.

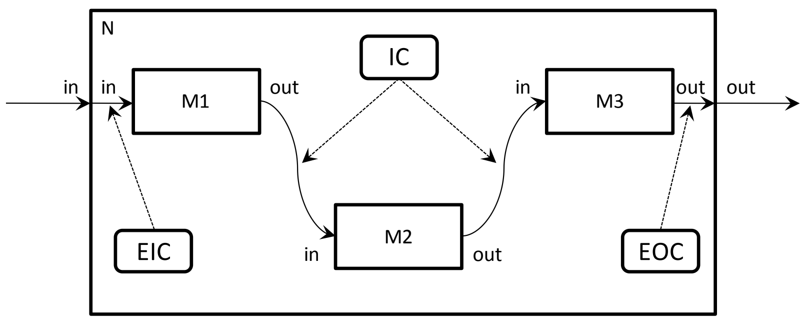

Figure 5: A coupled model showing different type of couplings

Understanding the dynamics of atomic and coupled DEVS system was essential to understand how DEVS and its extension, Dynamic Structure DEVS formally describes an adaptive system. It can now be seen that a DEVS system is a hierarchical complex dynamical system *closed under coupling* (similar to *closed under composition*) with modularity at its core. In its default description, while DEVS can specify a structurally static system, the formal atomic and coupled models can not sufficiently describe the network dynamics and adaptive behavior as needed for natural and biological systems, especially when stigmergy and emergence are occurring together. However, in its current state, it certainly can describe weak emergence in isolation. Examples such as an avalanche or a sand-pile are emergent phenomena with no self-organization. To model hierarchical self-organization, it is imperative to have a variable structure capability (Uhrmacher & Zeigler, 1996) that can reconfigure the component system both structurally and functionally. The structural capability is manifested externally, outside the component boundary, while the functional capability is manifested internally, within a component.

5.3 Dynamic Structure and Multi-Level DEVS

DEVS systems have a continuous time-base but their execution is event-based. A variable structure discrete event system adds a temporal nature to the structure of the system itself. The structure of a system can be dynamic at three levels:

1. *Component level*: entire sub-structures are removed or added in a live system
2. *Connection level*: interactions are reconfigured in a live system
3. *Interface level*: interface of the component itself is subject to reconfiguration

The behavior of a system can be dynamic in four ways:

1. State space
2. Time advances of each state
3. Transition functions ($eg.\ \delta_{int}, \delta_{ext}, \delta_{con}, \lambda$)
4. Initial state

More technically, such dynamism must be traceable to the levels of system specification described in Section 5.1. Table 3 provides the mapping of how dynamism is introduced at various levels. It shows what would be an outcome of such dynamic activity. The last column relates it to the list of questions we encountered in understanding the nature of scale-free networks.

Level	Name	How dynamism is introduced	Outcome	Impact in a Scale-free network
4	Coupled Systems	1. System substructure 2. System couplings 3. Subsystem I/O interfaces 4. Subsystem active/dormant	1. dynamic component structures 2. dynamic interaction	*II, IV, VII*
3	I/O System	1. Addition/removal of states 2. Augmentation of transitions with constraints/guard conditions	1. dynamic states 2. dynamic transitions 3. dynamic outputs	*I, III, IV, VI, VII*
2	I/O Function	1. Initial state 2. Addition/removal of initial state 3. Addition/removal of I/O pairs	dynamic initial state	*IV*
1	I/O Behavior	1. Time scale between the I/O behavior 2. I/O mapping changing the behavior itself 3. Allowed behavior 4. Addition/removal of I/O pairs	dynamic I/O behavior	*I, II, III,V,VII*
0	I/O Frame	1. Allowed values 2. I/O to port mapping	dynamic interfaces	*III, VII*

Table 3: Introducing dynamism at various levels of system specifications

The dynamic structure outcomes have been adequately dealt with in our earlier work (Hu, Zeigler, & Mittal, 2005), formally by Barros (1995; 1997; 1998), Uhrmacher (2001), Uhrmacher & Priami (2005) and Uhrmacher, et al (2006). Here we will discuss the formal undertaking of dynamic structure by Uhrmacher (2001) and Uhrmacher et al (2011) as the structural change is initiated from within the system components rather than a specialized component called Network Executive in Barros' DSDEVS. The first version is named as DynPDEVS. The underlying idea behind this DEVS extension is to interpret models as a set of models successively generating themselves by model transitions. Model and network transitions are introduced mapping the current state of a model into a set of models that the model belongs to. The formalism supports models that adapt their own interaction structure and their own behavior as a result of those interactions through a newly added transition function ρ_α. The structural changes are induced bottom-up and are communicated through another newly defined transition function, ρ_λ. Specific types of input and output interfaces are introduced that communicate these structural changes to other models. This version refers to dynamic components and dynamic coupling in a live system.

The second more advanced type is built on DynPDEVS and it introduced dynamic port interfaces. The ports X, and Y are part of the incarnations of model M. This is the most critical of capabilities requires for metamorphosis of the component allowing plasticity (Mittal et al 2005), for example, in neuronal ensembles that add dendrites and axons to support the Hebbian hypothesis. A DEVS neuron with dynamic interfaces requires this capability of dynamic interfaces as it strengthens or weakens its connections with other neurons. The second version is named ρDEVS. Formally, it is described as:

An atomic $\rho DEVS$ is a structure $< m_{init}, M, X_{sc}, Y_{sc} >$ with $m_{init} \in M$ being the initial model, and $m_i \in M$ be the i^{th} incarnation, X_{sc} and Y_{sc} the ports to communicate structural changes, and M the least set with the following structure:

$M_{\rho Devs} =< M_{pDevs} M_{pDevs}, s_0, \rho_\alpha, \rho_\lambda >$, where

$s_0 \in S$ *initial state*
$\rho_\alpha : S \times X_{sc} \rightarrow M$ *model transition function*
$\rho_\lambda : S \rightarrow Y_{sc}$ *implied structural network changes*

A reflective, higher order network, a ρ-NDEVS, is the structure $\rho NDEVS = < n_{init}, N, X_{sc}, Y_{sc} >$ with $n_{init} \in N$ being the start configuration, X_{sc} and Y_{sc} the ports to communicate structural changes, and N the least set with the following structure:

$N_{\rho DEVS} = < X, Y, C, MC, \rho_{N,} \rho_\lambda >$, where

C *set of components that are of type* $\rho DEVS$
MC *set of multicouplings*
$\rho_N : S^N \times X_{sc} \rightarrow N$ *network transition function*
$\rho_\lambda : S^N \rightarrow Y_{sc}$ *structural output function*

The value of ρ_N preserves the state and the structure of models that belong to the "old" and the "new" composition of the network. A multicoupling $mc_i \in MC$ in this formalism determines how the outputs are distributed from output to input ports. In regular DEVS (section 5.1), if more than one input port is linked to an output port, the output values are cloned at all the inports. When the artifacts and messages are in real world and consumable physical objects, this may not be desirable. The standard strategy is useful when the information is to be broadcast. In natural systems, the capability warrants a function that selects the output port for consumable resources. A random selection strategy may very well be used in the MC function. For rigorous mathematical analysis of this formalism, see Uhrmacher, et al. (2006).

The dynamic structure capability thus far defined by *ρDEVS* is manifested externally in the topology. An atomic model can be reincarnated as a coupled model and hierarchy can emerge. However, the coupled component still acts as a container of other components without any state and behavior representation. Hubs cannot form without displaying a behavior. To alleviate this problem, state and transition functions are introduced at the coupled level in Multi-Level-DEVS (ML-DEVS) (Uhrmacher, et al., 2007). ML-DEVS is an extension of *ρDEVS* and consists of Micro-DEVS (atomic) and Macro-DEVS (coupled). Let us look at Macro-DEVS first. A Macro-DEVS has structured input, output and state sets, *X*, *Y* and *S* respectively. An λ output function produces output for the output ports and a set *C* of components is specified. A set of multi-coupling functions *MC* allows specification of *value* couplings. The state transition function δ takes into account the current state, the components and multi-couplings to calculate the new state. A function *p* associates ports with each state. The structural change function *sc* defines the correlation between the set of components and multi-couplings for the current state. The downward causation is enabled by v_{down} that couples Macro-DEVS' current state variables to the input ports of micro-DEVS. The downward activation is done by λ_{down} function that allows synchronous activation of micro-DEVS models in an event-based manner. The upward causation is enabled by the port transition function as all the available ports at micro-DEVS level are available at macro-DEVS level. The transition function δ at Macro-DEVS accounts for any change in ports at the micro-DEVS in calculation of next Macro-DEVS state.

A Macro-DEVS is defined as a structure:

$$N_{mlDevs} =< X, Y, S, s_{init}, p, C, MC, \delta, \lambda_{down}, v_{down}, sc, act, \lambda, ta >, where$$

p:	*function that maps ports with each state s*
C:	*set of sub-models which are of type Micro-DEVS or Macro-DEVS*
MC:	*set of multi-couplings,* $\{m \mid m: 2^P \to 2^P\}$
$\delta: X \times Q \times 2^{C\times P} \to S$	*state transition function*
$\lambda_{down}: S \to 2^{Y\times C\times P}$	*downward output function*
$v_{down}: V_s \to P$	*value coupling downward*
$sc: S \to 2^C \times 2^{MC}$	*structural change function*
$act_{up}: S \times 2^{C\times P} \to \{true, false\}$	*activation function*

For more detailed mathematical analysis, see Uhrmacher, et al. (2007). The application of ML-DEVS has been in the areas of computational chemistry and biology. As a result, the formalism was designed to satisfy the needs of these disciplines where agents are essentially reactive. Micro-DEVS is a simplified version of *parallelDEVS* in which there is no δ_{int} and δ_{con}, but only δ_{ext} to account for external messages. This simplification is undesirable when the agent is proactive and adaptive with learning behavior. The agent's internal state is equally important and is much needed. Consequently, Micro-DEVS is unsuitable for modeling CAS. We recommend using the atomic *ρDEVS* for CAS. Macro-DEVS, being a coupled model, holds components, but also has state and various transition functions that enable upward and downward causation. Other examples in literature that deal with variable structure in multi-agent systems are Agent-Oriented DEVS (Uhrmacher & Zeigler, 1996). However, their atomic DEVS specification has to be integrated with Macro-DEVS to model the transformation of a node into a hierarchical node, i.e. a Hub.

Coming back to our discussion of CAS, let us now look at how the dynamic structure DEVS lends itself to describe a scale-free CAS.

6 DEVS for Complex Adaptive Systems

The feature list presented in Table 4 list just some of the features that we identified and that can help in modeling CAS with DEVS. Our analysis is based on scale-free topologies and co-occurrence of self-organization and emergence in an interconnected network of persistent agents and persistent environments. We also established that a stigmergic system is a type of CAS so the features categorized as "SG" in Table 4 are also applicable to CAS. The last column shows the state-of-the-art in modeling CAS and Stigmergy using Dynamic Structure Multi-Level DEVS.

ID	Feature	Category (SG/CAS)	What is answered?	Dynamic Structure DEVS	
				Can?	How?
A	Clustering	CAS	How does a node become a hub? How does the network handle hubs?	Yes / Partial	*ρDEVS and Macro-DEVS formalism together.* While the clustering can easily be implemented using value couplings, the transformation of a node into a hub and dynamic behavior of such transformation needs to be investigated
B	Scale-free topology	CAS	How does the network structures in presence of power law behave? How does the network connect nodes, clusters, and hubs in a scale-free topology?	Yes	*parallelDEVS* formalism. Co-occurrence of hubs and nodes with dynamic couplings and dynamic components
C	Preferential attachment	CAS	How does the new node in the network choose its neighbor based on affinity?	Yes	*ML-DEVS formalism* Value couplings allow development of contingency-based links that could reflect affinity and thresholds in a dynamic manner. One such framework called Knowledge-based Contingency Driven System (KCGS) framework (Douglass & Mittal, 2012) could specify the multi-level constraint network itself.
D	Growth and Decay	SG	How do the network linkages increase or decrease for a node?	Yes	*ρDEVS* formalism. Internal transition functions can direct inport and outport couplings along with dynamic component structures.
E	Threshold and Affinity	SG	How does the agent act upon various thresholds and how does it reconfigure its behavior?	Yes	*parallelDEVS* formalism. Transition functions can have threshold and affinity models
F	Inter-connectivity	SG	How is the dynamic nature of network is specified?	Yes	ρDEVS formalism

G	Modularity	SG	How does the external interface of an agent guide its role in network?	Yes	*parallelDEVS* formalism It is the very foundation of DEVS sytems
H	Hierarchy	CAS	How do clusters and hubs reduce their connectivity and change their role from a performer to an enabler?	Yes	*parallelDEVS* formalism DEVS complex systems are hierarchical by design.
I	Agent Persistence	SG	How does an agent handle persistent state? How is memory defined in an agent?	Yes	*parallelDEVS* formalism. Agents have state variables and are persistent. The state variables persist along the entire life cycle of the agent.
J	Environment Persistence	SG	How does an environment handle persistence? How do the affordances provided by the environment persist?	Yes	Loosely coupled, agent is modular and environment is external and unpredictable. Environment is available as an external activity through a Netcentric infrastructure. The agents developed in *parallelDEVS* as implemented using DEVS/SOA framework are loosely coupled with external web services through modular interfaces (Mittal & Martin, 2012).
K	Interactive Transition Systems	SG	How does an agent or a system specify its transition functions in an interactive manner?	Yes	*parallelDEVS* formalism. The three transition functions are based on a notion of abstract event that either triggers an internal transition or an external transition or both. A message exchange is an indication of an event at both the sender's and the receiver's end and is formally dealt with.
L	Self-organization	SG	How does an agent system organize itself towards a global behavior? How does it reconfigure its behavior?	Yes	ρDEVS *and ML-DEVS formalism* ρDEVS handles structural dynamism i.e. components, behavior and value couplings. ML-DEVS allows specification of constraints through value couplings that dictate coupling formation. Possible integration with KCGS framework may allow constraints specification (Douglass & Mittal, 2012).
M	Weak Emergence	SG	How does a system display global behavior greater than the behavior of its constituents?	Yes	*parallelDEVS* formalism. Emergence is an outcome. Specific observer agents can be coupled to the system who detect "emergent" parameters and activity.
N	Strong Emergence	SG	How does an observer embedded in a persistent agent, such that it reconfigures its external behavior, move to a higher level hierarchy to enable causal behavior at lower level?	Yes / Partial	*parallelDEVS* formalism. The observer is a DEVS agent that observes another DEVS agent or any external modular component. Such an observer can cause behavior change in the observed agent. A tightly coupled *agent+observer* coupled system becomes a *composite* agent with an embedded observer. A partial workable solution is thus provided.

O	Non-linearity	CAS	How does an event cascade in a network resulting in cascaded effects?	Yes	*Quantized-DEVS and ML-DEVS formalism* Value couplings communicate messages at various levels of hierarchy resulting macro-micro effects.
P	Concurrency	SG	Agent displays many parallel executing behaviors	Yes	*parallelDEVS*
Q	Upward Causation	CAS	How do the nodes in a hierarchical environment communicate information to hubs thereby eliciting reaction at a level above it?	Yes	*ML-DEVS*
R	Downward causation	CAS	How do the hubs cause changes at lower levels of hierarchy	Yes	*ML-DEVS*

Table 4: Features required for modeling scale-free CAS, capable of self-organization and emergence

All the feature requirements for stigmergy and some of the features of CAS are addressed by DEVS formalism. We can clearly see that clustering and strong emergence are the two properties that require augmentation to the current DEVS extensions. The clustering property specifically belongs to CAS and is not needed for modeling stigmergy. The current state of ML-DEVS is fully equipped to specify a stigmergic system except the partial solution provided for strong emergence. The current state of DEVS extensions is shown in Figure 6. Augmentation of the strong emergence capability, i.e. embedding the observer functionality inside an agent model will formally specify "Stigmergic-DEVS." Similarly, augmentation of the clustering capability, i.e. transformation of a node into a hub at both the structural and behavioral level, would specify CAS-DEVS. Since ML-DEVS is based on $\rho DEVS$, the ML-DEVS extension should be augmented to:

1. Transform an atomic component to a cluster component: this requires addition and augmentation of new transition functions in a live system such that it performs a macro-role rather than a micro-role. This is related to rise in abstraction at the DEVS atomic level. Augmentation should result in an algorithm that transforms a node into a hierarchical node with Macro-DEVS behavior.
2. Strong emergence: this capability requires the agent to reconfigure its behavior based on its observation of the micro- and macro- patterns in a downward causal manner as designed by the designer of the artificial system.

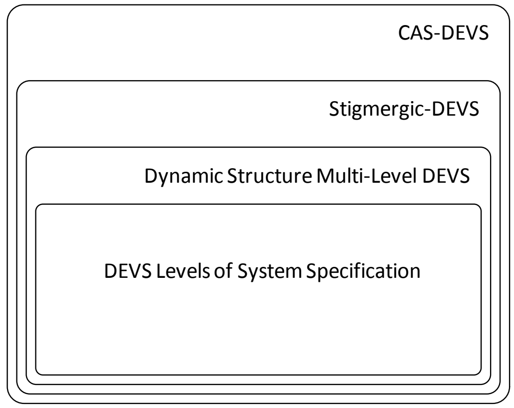

Figure 6: Stigmergic DEVS as an extension of Dynamic Structure DEVS

6.1 Discussion

Features listed in Table 4 operate at various levels of abstractions and an implementation of these features at the appropriate level of abstraction yields the desired effect. Next, we associate these features with DEVS Levels of system (Table 5). The presence of the same feature at different levels of DEVS specification implies that the feature needs to be implemented at all those levels. For example, feature A should be specified at levels 1, 2, 3 and 4 simultaneously to get the clustering effect. We introduced a "coupling" abstraction level in the coupled system at Level 4 to clearly mark the features that impact connectivity of atomic and coupled components. This may imply that there is an additional level of abstraction between the DEVS atomic and coupled components that formally specifies a dynamic coupling relation. The dynamic coupling relation has been described with reference to *ρDEVS* and *ML-DEVS.* As can be seen from Table 5, a coupled system at Level 4 is mirroring the feature set of an atomic system at Level 3 with the exception of features of hierarchy (including containment) and environment persistence. This also reaffirms our thesis that a coupled model specification needs to have a behavior of its own and not just act as a container. Further, the coupling abstraction may also cater to features like clustering, topology, preferential attachment, growth and decay, interconnectivity and self-organization. This implies:

1. That there may be a way to formally define a "rich" coupling specification that has above-mentioned aspects encoded.
2. That some of the behavior encoded in the nodes can become the behaviors of the networked system where the formal coupling specification manifests the properties of the complex network

Level	Name	Features of scale-free CAS
4	Coupled System	A, B, C, D, E, G, H, I, J, K, L, M, N, P, Q, R
	Couplings	A, B, C, D, F, L
3	I/O System	A, B, C, D, E, G, I, K, L, M, N, O, P, Q, R
2	I/O Function	A, C, D, F, I, N, Q, R
1	I/O Behavior	A, C, E, F, H, I, K, O, L, M, N, Q, R
0	I/O Frame	F, G, K, N, L, O, Q, R

Table 5: Abstraction levels of scale-free CAS features portraying self-organization and emergence

Another important aspect warranting discussion is the *closure under coupling* property of complex systems. The DEVS levels of system specification are *closed under coupling*, i.e. the behavior of a coupled DEVS can be specified as an atomic DEVS. This property helps build hierarchical complex systems and the current DEVS formalism is positioned to support weak emergence, whereby the emergent behavior can be reduced to lower level behavior of the constituent components of the system. In order to display strong emergence, what is needed is an extension of "closure under coupling" property of CAS such that the novel emergent behavior that is irreducible to the constituent components can be accommodated. This implies that the new observed behaviors (or emergents) that are not part of the system (at Level L_{n-1}) be made available as observers at a higher level of hierarchy at Level L_n, become "acquired" behaviors at Level L_n. Such acquired behavior should then reconfigure the Macro-DEVS behavior specification to incorporate the new abstraction and concepts as provided by the observers at L_n. Formal analysis of strong emergence and the corresponding *closure under coupling* property in DEVS CAS formalism is left for future work.

7 Conclusions

Complexity is a multifaceted topic and each complex system has its own properties. However, some of the properties like high interconnectedness, large number of components, and adaptive behavior are present in most natural complex systems. We looked at the mechanism behind interconnectedness using network science that describes many natural systems in the light of power laws and self-similar scale-free topologies. Such scale-free topologies bring their own inherent properties to the complex system such that the entire system is subjected to the network's structural and functional affordances.

It is largely unknown what makes a network evolve into a scale-free network, whether it is a top-down goal-driven phenomena or bottom-up causation or just an outcome of natural interactions. Two conditions have to be present for a network to evolve into a scale-free network: 1. incremental growth and 2. preferential attachment. We explored the notions of scale-free nature, strong and weak emergence, self-organization and stigmergic behavior in a complex adaptive system with persistent agents and persistent environment. We also related the concept of emergence to network science and presented arguments on how hubs and connectors are formed when a complex system is going through a critical phase. We argued that under any occurrence of both self-organized and emergent behavior together, the properties of scale-free network exist and one has to look at right level of abstraction in a multi-level system to witness the instance based interactions. We established that stigmergy displays strong emergence and is a specialized case of CAS. We also enumerated 18 properties of a CAS, 11 of which were properties of stigmergic systems.

We presented a high level view of DEVS theory and how its formal rigor is able to specify complex hierarchical systems. We described variants of dynamic structure and multi-level DEVS, and mapped it to some of the identified properties of CAS and stigmergy. We detailed the adaptive nature of complex system with DEVS Level of system specification and what it means to have dynamic adaptive behavior at different levels of a system. During the mapping process, we found that the following capabilities warrant formal attention to extend DEVS theory of complex systems to a theory of complex adaptive systems:

1. How clusters are formed, hubs appear and evolve.
2. How multi-level self-organization occurs.
3. How strong emergence results in self-organization with an embedded observer capable of causal behavior at lower levels of hierarchy.
4. How formal attention to coupling specification may provide additional abstraction mechanisms to model dynamic interconnected environment.

Finally, we recommended the augmentation of $\rho DEVS$ as the foundation for Stigmergic-DEVS, and investigation of both $\rho DEVS$ and ML-DEVS augmented together as a foundation for CAS-DEVS.

Acknowledgements

We would like to thank Margery J. Doyle for her expert knowledge on agent-based stigmergic behavior. We also thank reviewers for making insightful comments thereby increasing the quality of the article. Lastly, we thank our editors Leslie Marsh and Margery J. Doyle for their help in preparation of the manuscript.

References

Altman, R. C. (2009). Altman - The great crash. *Foreign Affairs.* Council on Foreign Relations, Retrieved May 29, 2012 from http://www.foreignaffairs.com/articles/63714/roger-c-altman/the-great-crash-2008#

Baas, N. (1994). Emergence, hierarchies and hyperstructures. *Artificial Life, 3*.

Banabeau, E., & Dessalles, J. (1997). Detection and emergence. *Intellica, 2*(25).

Barabási, A.-L. (2003). *Linked: How everything is connected to everything else and what it means for business, science and everyday life. New York, NY,* Penguin Books.

Barros, F. (1995). Dynamic structure discrete event system specifications. *Proceedings of the 1995 Winter Simulation Conference.* Arlington, VA.

Barros, F. (1997). Modeling formalisms for dynamic structure systems. *ACM Transactions on Modeling and Computer Simulation, 7*(4): 501-515.

Barros, F. (1998). Abstract simulators for the DSDE formalism. *Proceedings of the 1998 Winter Simulation Conference.* Washington, DC.

Bilke, S., & Peterson, C. (2001). Topological properties of citation and metabolic networks. *Physical Review, E*(64).

Bradonjic, M., Hagberg, A., & Percus, A. G. (2007). Giant component and connectivity in geographical threshold graphs. Proceedings of the 5th International Conference on Algorithms and Models for the Web-graph, (pp. 209-216). Springer-Verlag Berlin, Heidelberg

Brooks, R. (1991). Intelligence without reason. *12th Int'l Joint Conference on Artificial Intelligence.* San Mateo, CA.

Buchanan, M. (2001). *Ubiquity: the science of history ... or why the world is simpler than we think.* New York: Crown Publishers.

Camazine, S., Deneubourg, J., Franks, N., Sneyd, N., Theraulaz, G., & Bonabeau, E. (2001). *Self-Organization in biological systems.* Princeton: Princeton University Press.

Chalmers, D. J. (2006). Strong and weak emergence. In *The re-emergence of emergence.* Oxford: Oxford University Press.

Deguet, J., Demazeau, Y., & Magnin, L. (2006). Elements about the emergence issue: A survey of emergence definitions. . *Complexus, 3: 24-31.*

Douglass, S., & Mittal, S. (2012). A framework for modeling and simulation of the artificial . In *Ontology, epistemology and telelogy* (Ed. A. Tolk). Philosophical Foundations for Intelligent M&S Applications Series. Springer USA.

Doyle, M., & Kalish, M. (2004). Stigmergy: Indirect communication in multiple mobile autonomous agents. In *Knowledge systems for coalition operation* (pp. 151-158). Prague: Czech Technical University.

Goldin, D., Smolka, S.A. & Wegner, P. (2001). Turing machines, transition systems, and interaction. *8th Int'l Workshop on Expressiveness and Concurrency.* Aarlbog, Denmark.
Golding, D. (2000). Persistent turing machines as a model of interactive computation. In *Proceedings of the First International Symposium on Foundations of Information and Knowledge Systems.* Cottbus, Germany.
Grassé, P. P. (1959). La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie : Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6(1), 41–83.
Heyligen, F. (2002). The science of self-organization and adaptivity. In *The Encyclopedia of Life Support Systems.* Unesco Publishing-Eolss Publishers.
Holland, J. H. (1992). Complex adaptive systems. *Daedalus, 121*(1): 17-30.
Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity, 19*(1): 1-8.
Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life, 5*(2): 173-202.
Hu, X., Zeigler, B., & Mittal, S. (2005). Variable structure in DEVS component-based modeling and simulation. *Transactions of SCS, 81*(2): 91-102.
IMF (2009). World Economic Outlook: A Survey by the Staff of the International Monetary Fund. Washington, DC.
Izquierdo-Torres, E. (2004). *Collective intelligence in multi-agent robotics: Stigmergy, self-organization and evolution.* University of Sussex, UK
Jeong, H., Mason, S., Barabási, A. & Oltvai, Z. (2001). Centrality and lethality of protein networks. *Nature,* 411: 41-42.
Jeong, H., Neda, Z., & Barabsi, A. (2003). Measuring preferential attachment for evolving networks. *European Physics Letters,* 61: 567-572.
Jeong, H., Tombor, B., Albert, R., Oltvai, Z. & Barabási, A. (2000). The large-scale organization of metabolic networks. *Nature,* 407: 651-654.
Kadanoff, L. P. (1993). *From order to chaos* (Essays: Critical, chaotic and otherwise). Singapore: World Scientific.
Keil, D., & Goldin, D. (2003). Modeling indirect interaction in open computational systems. *12th IEEE Int'l Conference on Enabling Technologies: Intrastructure for Collaborative Enterprises.*
Milgram, S. (1967). The small world problem. *Psychology Today*. 1(1): 60-67.
Mittal, S. & Martin, J.L., (2012). *Netcentric system of systems engineering with DEVS unified process*, Boca Raton, FL: CRC Press
Mittal, S., Khargharia, B. & Zeigler, B. (2005). *Neuronal path formation using spiked leaky neuron model.* Tucson, AZ: Univesity of Arizona, ACIMS Lab, ECE 549 project.
Muller, J. (2003). Emergence of collective behavior and problem Solving. *4th International Workshop on Engineering Societies in the Agents World. London, UK*
Newman, M. (2001). Clustering and preferential attachment in Growing Networks. *Physical Review, E 64*(2), 025102
Newman, M. (2005). Power laws, pareto distributions and Zipf's law. *Contemporary Physis,* 46(5): 323-351.
Newman, M., & Watts, D. (1999). Renormalization group analysis of the small-world network model. *Physics Letters,* A(263): 341-346.
Pinker, S. (1997). *How the mind works.* New York: WW. Norton.
Resnick, M. (1994). *Turtles, termites and traffic jams.* Cambridge, MA: MIT Press.
Shalizi, C. (2001). *Causal architecture, complexity and self-organization in time series and cellular automata.* Madison: PhD Thesis, University of Wisconsin at Madison.
Siegelmann, H. (1999). *Neural networks and analog computation: Beyond the Turing limit.* Boston, MA. Birkhauser.
Stanly, H.E. (1971). *Introduction to phase transitions and critical phenomena*. Oxford, England: Oxford University Press
Sulis, W. (1997). Fundamental concepts of collective intelligence. *Nonlinear Dynamics, Pscychology, and Life Sciences, 1*(1):35-53
Treasury, H. (2007). *News release: Liquidity support facility for Northern Rock plc.* Bank of England and Financial Services Authority.
Uhrmacher, A. (2001). Dynamic structures in modeling and simulation - A reflective approach. *ACM Transactions on Modeling and Simulation,* 11(2): 206-32.
Uhrmacher, A. M., & Zeigler, B. P. (1996). Variable structure models in object-oriented simulation. *International Journal of General Systems,* 24(4): 359-375.

Uhrmacher, A. & Priami, C. (2005). Discrete event systems specification in systems biology - a discussion of stochastic pi calculus and DEVS. *Winter Simulation Conference.* Orlando, FL.

Uhrmacher, A., Ewald, R., John, M., Maus, C., Jeschke, M., & Biermann, S. (2007). Combining micro and macro-modeling in DEVS computational Biology. *Winter Simulation Conference*. Washington, DC

Uhrmacher, A., Himmelspach, J., & Ewald, R. (2011). Effective and efficient modeling and simulation with DEVS variants. In *Discret event modeling and simulation: Theory and applications* (pp. 139-176). Boca Raton, FL. CRC Press.

Uhrmacher, A., Himmelspach, J., Rohl, M., & Ewald, R. (2006). Introducing variable ports and multi-couplings for cell biologinal modeling in DEVS. *Winter Simulation Conference*. Monterey, CA

Vangheluwe, H. (2000). DEVS as a common denominator for multi-formalism hybrid systems modeling. *IEEE International Symposium on Computer Aided Control System Design.* IEEE Computer Society Press. 134:129-134

Watts, D. (1999). *Small worlds.* Princeton: Princeton University Press.

Watts, D., & Strogatz, S. (1998). Collective dynamics of small-world networks. *Nature* (393): 440-442.

Wilson, K. G. (1971). Renormalization group and critical phenomena: I. Renormalization group and the Kadanoff scaling picture. *Physics Review,* B(4): 3174-3183.

Wolf, T. D., & Holvoet, T. (2005). Emergence versus self-organization: Different concepts but promising when combined. *Lecture Notes in Artificial Intelligence,* 3464: 1-15.

Zeigler, B.P. (1976). *Theory of modeling and simulation*. Wiley Interscience

Zeigler, B. P. (2004). Discrete event abstraction: An emerging paradigm for modeling complex adaptive systems. In *Perspectives on Adaptation in Natural and Artificial Systems, Essays in Honor of John Holland.* Oxford: Oxford University Press.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). *Theory of modeling and simulation* (2nd ed.). San Diego, CA. Academic Press.