A Pragmatic Complexity Framework

[1] System Concepts, USA, jjs-sbw@eskimo.com
[2] System Concepts, USA, mjs-sbw@eskimo.com

Abstract
Complexity and complex systems are two recurrent themes that are deeply woven into the literature, culture and practice of systems thinking, systems science and systems engineering. This paper presents an outline for a pragmatic complexity framework that establishes a rule set to facilitate the clear identification, organization, and communication of system and complex system attributes and characteristics. Specific conceptual and process techniques are presented that can be used to reduce the relative complexity of a given system. A number of classical systems engineering analysis and evaluation problems have been selected from the literature and have been evaluated, analyzed and addressed using the operations associated with the proposed pragmatic complexity framework.

Keywords – Complexity, Complex Systems, Cognitive Complexity, Abstract Relation Type, System Patterns, N-Squared Chart, Design Structure Matrix, Systems Engineering.

1 Introduction

The pragmatic complexity framework establishes a uniform set of rules that are used to clearly identify and communicate the attributes of each defined complexity type as it relates to a specific context. Constructed of fundamental layers of functional nodes and links, the pragmatic complexity framework details the source, interface and expression point of each type of complexity. As reported in this research, various types of complexity can be combined to produce complexity of a completely different type and nature. The primary design goal of the pragmatic complexity framework is to establish a standard set of techniques and operations that may be used to reduce complexity in the organized tasks associated with large scale systems and systems of systems design. Based on the long standing system engineering tradition of the use of N-Squared Charts as well as graphical techniques from the systems dynamics community, this research presents a language of complexity that has equivalent graphical, mathematical and prose forms. The rest of the paper is organized into sections addressing definitions, assumptions, related work, related process concepts, classical techniques, current techniques, a pragmatic complexity framework outline, as well as summary and future work. The examples presented here detail the application of the outlined techniques as they are applied to classical N-Squared Charts, design structure matrix diagrams, and other system analysis techniques that use a matrix representation of the system and/or system structure.

2 Definitions

Terms that are important to the discussion of complexity and complex systems engineering are presented here along with the source of the definition to help establish a semantic basis to support the following discussion.

2.1 System

"A system is a set of objects together with relationships between the objects and between their attributes [1]." Five basic definitions of a system are given by Klir [2]: 1)" The system S is a given set of quantities regarded at a certain resolution level". 2) "The system S is a given ensemble of variations in time of some quantities under consideration." 3) "The system S is a given time-invariant relation between instantaneous and/or past and/or future values of external quantities; every element of the relation may, but need not, be associated with the probability of its occurrence." 4) "The system S is a given set of elements, together with their permanent behaviors, and a set of couplings between the elements on the one hand, and between the elements and the environment on the other." 5) "The system S is a given set of states together with a set of transitions between the states; every state may, but need not, be associated with a probability of its occurrence." Two general and two mathematical definitions of a system are provided by Warfield [3]: General 1) "A complex unity formed of many, often diverse, parts subject to a common plan or serving a common purpose." General 2) A composite of equipment, skills, and techniques capable of performing and/or supporting an operation. A complete system includes related facilities, equipment, material, services, and personnel required for its operation to the degree that it can be considered a self-sufficient unit in its intended operational (or non-operational) and/or support environment. Mathematical 1) "A set of quantities having some common property, such as the system of even integers, the system of lines passing through the origin, etc." Mathematical 2) A set of principles concerned with a central objective, as a coordinate system, a system of notation, etc. Two basic system definitions are provided by Steward [4]: 1) "A system is a thing, which if you make any change to it, there are likely to be many subtle consequences." 2) "A system is a collection of parts and relations between the parts such that the behavior of the whole is a function not only of the behaviors of the parts, but also of the relations among them." One system definition is provided by Koth [5]: "A system most generally implies the idea of elements forming an ordered whole, with the relations among these elements forming the structure of the system. The systemic character appears only as the elements function within the system." The following system definition is provided by Simpson and Simpson [6]: "A system is defined as a non-empty set of objects and a non-empty set of relationships mapped over these objects and their attributes."

2.2 Complexity

Warfield [7] first defines complexity as: "Complexity is a property of a system, occasioned by a variety of factors germane to the system, which complicate human perception of the system and consequently make its understanding difficult." Steward [4] defines complexity as: "Given the parts and their behaviors, complexity is the difficulty involved in using the relations among the parts to infer the behavior of the whole." Casti [8] defines complexity as "we arrive at the relativistic view of system complexity … Here we have natural system N being observed by the system O. The two arrows … represent the complexity of O as seen by N and the complexity of N as seem by O, using whatever measure of complexity one chooses. The important point here is that the complexity of a given system is *always* determined by some other system with which the given system operates." Bar-Yam [9] defines complexity as: "Complexity is a measure of the inherent difficulty to achieve a desired understanding. Simply stated, the complexity of a system is the amount of information necessary to describe it." Warfield [10] again defines complexity, in 2003, as "Complexity is that sensation experienced in the human mind when, in observing or considering a system, frustration arises from lack of comprehension of what is being explored." Complexity is a measure of the difficulty and/or effort and/or resources required for one system to effectively observe and/or interoperate with another system.

2.3 Computational Complexity

Computational complexity is associated with well defined algorithmic problems and the efficient solutions for these stated algorithmic problems.

2.4 Cognitive Complexity

Cognitive and/or perceptual complexity is associated with how difficult it is for a human to clearly understand the system problem and/or current situation.

2.5 Relationship

The term relationship is used to express the common idea used in informal speech that associates two or more objects in some manner.

2.6 Relation

The term relation is used to express the formal mathematical association between and among objects.

2.7 Structural Modeling

"Structural modeling employs graphics and words in carefully defined patterns to portray the structure of a complex issue, a system, or of a field of study. These patterns, called structural models, differ from ad hoc representations in that specific construction techniques and rules for their formation are founded on substantive theory." [7]

3 Assumptions

Many of the major assumptions associated with the development of a pragmatic complexity framework are stated in this section. These assumptions are either taken directly from *Structuring Complex Systems*, by Warfield [7], or are based on the assumptions and concepts presented in that work. These assumptions and concepts are considered central to the discussion and development of techniques that reduce the complexity associated with the analysis, design, communication, development and deployment of complex systems.

Assumption 1: "One major factor in complexity is the structural aspect of the system." [7]

Assumption 2: "Perception of a system structure becomes particularly difficult when the system includes many elements and relationships due, in part, to the combinatorial limitation of the human mind which, according to psychological research, precludes both simultaneous consideration of many elements and relationships and behavioural consistency in coping with them." [7]

Assumption 3: "Modelling is a useful activity to gain increased comprehension of a system, and since all modelling involves system structure explicitly or implicitly, a more powerful capability to model system architecture will be quite useful in probing complexity." [7]

Assumption 4: Computer-based methods are required to assist and enable the human mind to effectively cope with complexity.

Assumption 5: The combinatorial features of complex systems, coupled with the need for computer-based assistance, create the need for a mathematics that supports and facilitates the development and understanding of a system structure.

Assumption 6: "The development of system structure is inherently an iterative process, and the most inhibiting part of this process is the development of the first respectable approximation to the structure of a system." [7]

Assumption7: The availability and power of computing systems is much greater now, in 2009, than in 1974.

Assumption 8: Primary features of structural modelling techniques can be used to augment and improve a variety of classical system analysis and evaluation methods.

4 Related Work

There is a large body of system science, system analysis, system dynamics and system engineering work that has either a direct or indirect relationship to the work presented in this paper. The foundational related work is represented by the techniques of Interpretive Structural Modeling, Interactive Management [11], and Generic Design Science [12] all created by Warfield. Classical work in the area of Interpretive Structural Modeling and other techniques to address large-scale systems was reported by Sage [13] in, *Methodology for Large Scale Systems*, published in 1977. Techniques for systems analysis and evaluation that are closely associated with the current work were published by Steward [4] in 1981. Specific examples of system analysis and evaluation techniques presented by Lano [14] in *A Technique for Software and Systems Design*, will be addressed in later sections of this paper. Design structure matrix techniques, related to the work of Steward [4] and Warfield [11], will also be addressed in a later section of the paper using a detailed evaluation of published work. Other areas of related work are the dynamic and automated N-Squared Charts and the process for minimizing N-Squared Chart entropy that are contained in *Advanced Systems Thinking, Engineering and Management*, by Hitchins [15]. The final areas of related work are the papers published by Simpson that relate to the development and utilization of Abstract Relation Types [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27].

5 Related Process Concepts

In general, the processes presented here will be focused on the reduction of cognitive and/or perceptual complexity. Due to the fact that computer assistance will be required to augment the human perception and interface with the system structuring and complexity reduction activities, there may be an opportunity to confuse the different types of complexity associated with the system structuring and analysis activity. Therefore a basic relationship diagram is presented in Figure 1 that is used to clearly define the concept of complexity and its component parts. Any aspect of complexity that can not be completely associated with either the cognitive complexity component or the computational complexity component is assigned to the everything-else component of complexity. Cognitive complexity reduction techniques are emphasized in this paper, computational complexity reduction aspects of this work will be addressed in future papers. The separation of global system relationships from the local interface value set and/or sets provides the foundation necessary to address cogitative complexity at the global level, and computational complexity at the local level. In fact, this approach directly addresses Steward's definition of complexity in a pragmatic manner.

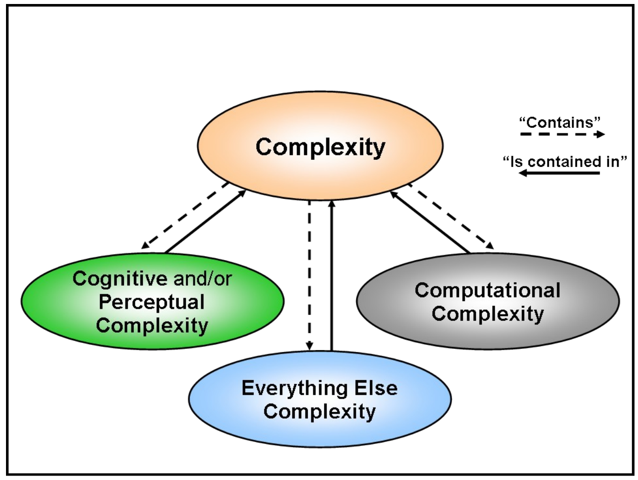

Figure 1 –*Complexity Components.*

Cognitive complexity reduction is achieved in general by understanding the limits associated with human cognition both as individuals and as groups. This understanding is then used to design methods of system analysis, evaluation and communication that use these limits as design constraints. The *Science of Generic Design* [12] addresses these human complexity limits using specifically designed language, mathematical and graphical constructs. A key concept used in cognitive complexity reduction is the tight binding of natural language "relationships" to mathematical relations. This strong semantic binding provides the ability for humans to explore the structure and context of complex systems using their natural language. The natural language semantics are then encoded into mathematical relations and computer executable processes that enable the computer based processes to augment human insight, understanding and perception of the given complex system. In essence, using the Science of Generic Design allows people to do what people do very well (and computers do poorly), and computers to do what computers do very well (and what people do poorly). The primary goal of this paper is to present computer-based, evolutionary algorithms that assist in the reduction of complexity by creating groups and clusters of system components that are combined into one system element. In this specific manner, the number of system components and interaction nodes are reduced, thereby directly reducing the cognitive complexity associated with given system representation and communication.

6 Classical Techniques

Two kinds of classical systems analysis techniques that have a graphical matrix component will be discussed and evaluated in this section. The first technique is the N-Squared Chart which is defined as [14], "The N-Squared Chart is a fixed-format structure which graphically displays the total bidirectional interrelationships between the individual functions and/or components within a given system or structure. The N-Squared Chart provides a structured method for the definition of functional interactions and interfaces. The chart itself is a graphical presentation of all of the functions within a system (subsystem, task, etc.), together with the one-way interactions between each of these functions ordered in a fixed coordinate matrix format.” The N-Squared Chart technique has greatly increased the ability of individuals and groups to organize, communicate and comprehend information and data associated with complex systems, system design and system operations.

Two basic examples taken from pages 24 and 25 in Section 2 of *A Technique for Software and Systems Design* [14] were selected for detailed evaluation and analysis. The first example was selected because the author reported that the analysis process produced a system that was simpler to define, control and test. In essence the author reported a reduction in system design and process cognitive complexity. The second example was selected because of the demonstrated reduction of links, nodes and symbols that were again associated with a decrease in complexity and an increase in the ease of system design and production.

Genetic algorithms and evolutionary computation have been used to effectively analyze and evaluate system matrix structures [23] [24] [25] [26]. The first selected example has been analyzed using evolutionary computational techniques and the published analytical result was duplicated. This result provides a positive indication that evolutionary computation can be applied as a useful analytical tool that reduces the human effort required for system functional and structural analysis while at the same time producing a system product that measurably reduces the cognitive complexity associated with the system representation. Before the evolutionary computation techniques could be applied to these problems some thought and effort was required to produce an operational system structuring technique. Starting with the most basic definition of a system, the contextual structuring relationship that was selected for use was “is connected to.” While the standard N-Squared Chart technique focuses on local interface relationships, there is no stated global organizing system relationship associated with the classical approach. The “is connected to” contextual relationship was selected because it is a natural language, transitive relationship that “binds” all of the function nodes on the chart into one global system structure. Using the Abstract Relation Type concept and approach, the binary matrix set was developed with a marking space matrix, one value space matrix and one outcome space matrix. The marking space matrix represents the system structure using a matrix that encodes a binary relationship, in this case, “is connected to.” If node *a* is connected to node *b*, then element (*a,b*) of the matrix contains a 1. If node *b* is connected to node *a*, then element (*b,a*) of the matrix contains a 1. Otherwise each element of the matrix contains a 0. Nodes cannot be connected to themselves so all of the marking space matrix diagonal elements contain a 0. The value space matrix set is used to represent the value associated with any specific system structural configuration represented by the marking space. The outcome space represents the combination and/or evaluations of the value space set as they relate to the system structure represented by the marking space. Figure 2 provides an overview of the ART binary matrix components.

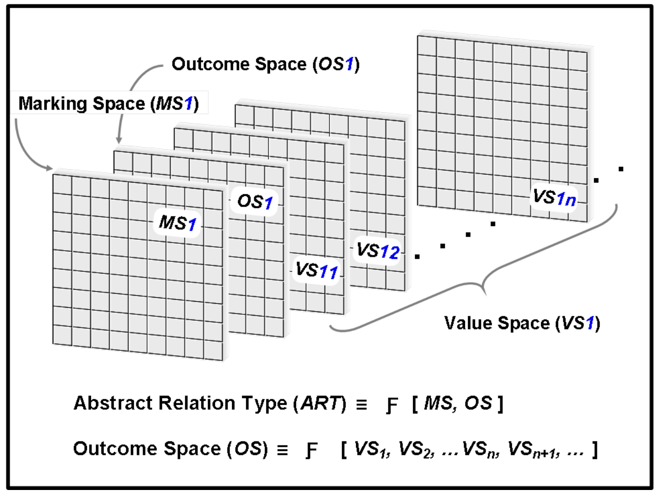

Figure 2 - *Abstract Relation Type (ART) Components.*

The value space selected for this evolutionary computation application consists of an increasing function starting with 0 (zero) at the matrix diagonal and increasing by 1 (one) for each matrix element away from the diagonal. The goal (or fitness function) of the evolutionary computational approach is to minimize the sum of the values in the outcome space. In this case the evolutionary computational fitness function could be a maximization function, but the value space would need to be changed to begin with a 7 (seven) at the diagonal and then the numbers would decrease moving away from the diagonal. Figure 3 presents Lano’s graphical functional decompositions [14] that have been adapted for use in this paper.

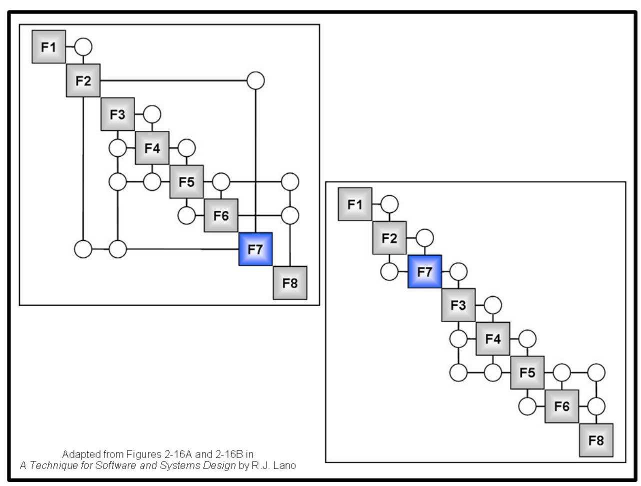

Figure 3 – *Initial and Final Functional Organization.*

Starting with a system configuration that has a value of 28, the evolutionary computational approach searches the solution space to find a minimum answer of 15, which represents the same system configuration produced and reported by the human systems analysts. The value space matrix is presented in Figure 4, while the initial and final marking space configurations are given in Figure 5.

Value Space 1 for Figures 2-16A and 2-16B

0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6
2	1	0	1	2	3	4	5
3	2	1	0	1	2	3	4
4	3	2	1	0	1	2	3
5	4	3	2	1	0	1	2
6	5	4	3	2	1	0	1
7	6	5	4	3	2	1	0

Figure 4 – *Minimization Value Space Arrangement.*

Initial Marking Space for Figure 2-16A

0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	1	0	1
0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Final Marking Space for Figure 2-16B

0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0
0	0	0	1	1	0	1	1
0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0

Figure 5 – *Initial and Final Marking Space Configurations.*

The second set of system representations selected for evaluation detail the reduction in cognitive complexity achieved by grouping the components of a system that has been subjected to detailed functional analysis. The original functional diagram contains 9 (nine) functions and 23 (twenty three) interaction nodes. The final functional implementation diagram contains 5 (five) functions and 8 (eight) interaction nodes. The final diagram represents a 44 percent reduction in the number of functions and a 65 percent reduction in the number of interaction nodes. The magnitude of the reductions may not be as important as the fact that the final numbers are within the short term memory capacity of human being as presented by Miller [27]. The complexity limit for human short term memory is given as 7 (seven) items plus or minus 2 (two). It is clear that both the number of functions, 5 (five), and the number of interaction nodes 8 (eight) in the final implementation diagram are within the Miller short term memory limit. Figure 6 presents an adaptation of the Figures 2-17 A and 2-17B [14]. The author points out that both graphical representations represent the same system and an operational system could have been constructed using either system representation. However, the system representation with the smaller number of functions and nodes was easier to communicate and present across the group of individuals from different disciplines that were required to work together to implement the final system product.

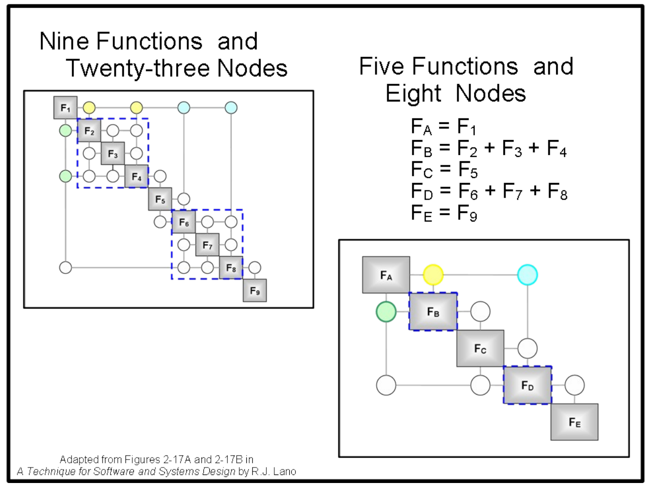

Figure 6 – *Cognitive Complexity Reduction Example.*

The second set of classical techniques that will be evaluated are taken from the system analysis and management work of Steward as they are presented in his book, *Systems Analysis and Management Structure, Strategy, Design* [4]. Steward's initial work that instigated his study of system and systems structure was focused on the efficient solution of a set of linear equations using a computer. If the computing approach was properly arranged the solution to the problem could be attained in a much shorter time. This computing activity characteristic places the initial work solidly in the area of computational complexity reduction. However, as Steward's work progressed, methods and techniques were developed to enhance the understanding and communication of system and systems structure using graphical techniques. The graphical enhancement of understanding and communication among a design team places the work solidly in the area of cognitive and/or perceptual complexity reduction. Computer-based techniques that are fundamental to the reduction of cognitive complexity may also have a strong tie to computational complexity techniques. The main motivating reason to develop the pragmatic complexity framework is to provide a clear set of distinctions that can be used to identify and communicate detailed information about these different types of complexity as they apply to the many analytical tasks associated with systems science, systems thinking and systems engineering.

The systems analysis work is roughly divided into working with aspects of the system structure and as well as aspects of the system semantics. Steward [4] presents the analysis techniques for the structural analysis of many types of systems that have a wide range of system semantics, contextual semantics and application semantics. Examples are provided for the design of an electric car, the analysis of a national economic model, design of critical path schedules as well as an analysis of factors impacting the investment in cities. The electric car design problem that is presented in Steward's Section 2.6 will be considered in this section of the paper. An evolutionary computational approach was used to structure the given example and the reported published results were replicated using the evolutionary computational methods associated with the ART approach. Figure 7 presents the initial and final ART marking spaces produced by the evolutionary computational approach. Figure 8 is the ART value space associated with the electric car problem.

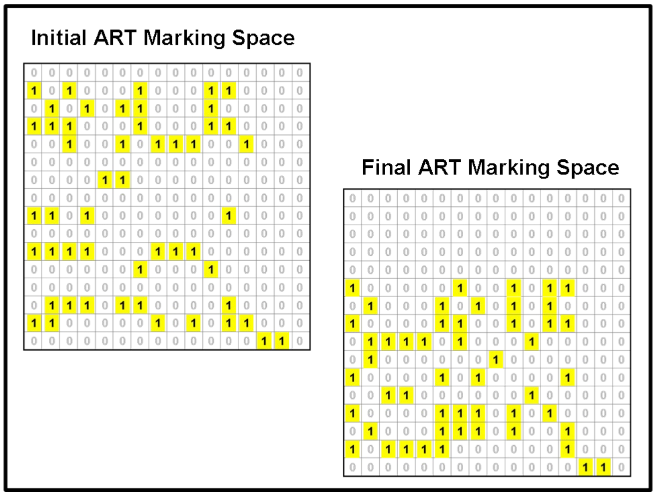

Figure 7 – *Initial and Final Electric Car ART Marking Spaces.*

0	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73
29	0	58	59	60	61	62	63	64	65	66	67	68	69	70	71
28	27	0	57	58	59	60	61	62	63	64	65	66	67	68	69
27	26	25	0	56	57	58	59	60	61	62	63	64	65	66	67
26	25	24	23	0	55	56	57	58	59	60	61	62	63	64	65
25	24	23	22	21	0	54	55	56	57	58	59	60	61	62	63
24	23	22	21	20	19	0	53	54	55	56	57	58	59	60	61
23	22	21	20	19	18	17	0	52	53	54	55	56	57	58	59
22	21	20	19	18	17	16	15	0	51	52	53	54	55	56	57
21	20	19	18	17	16	15	14	13	0	50	51	52	53	54	55
20	19	18	17	16	15	14	13	12	11	0	49	50	51	52	53
19	18	17	16	15	14	13	12	11	10	9	0	48	49	50	51
18	17	16	15	14	13	12	11	10	9	8	7	0	47	48	49
17	16	15	14	13	12	11	10	9	8	7	6	5	0	46	47
16	15	14	13	12	11	10	9	8	7	6	5	4	3	0	45
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 8 – ***Electric Car ART Value Space.***

7 Current Techniques

This section focuses on two areas: the automated N-Squared Chart approach presented by Hitchins [15] and the Design Structure Matrix approach presented by Sharman and Yassine [28]. This work has been addressed before by Simpson et. al. [21] [22] [23], however, this paper will expand the scope and detail associated with the discussion of these system analysis approaches. Hitchins [15] in 'Part II Systems Thinking,' of his book, *Advanced Systems Thinking, Engineering and Management*, introduces and explains the classical use of N-Squared Charts along with techniques that have been developed to integrate systems represented by these charts using encapsulation and elaboration techniques. N-Squared Charts are further developed by introducing the idea of dynamic and automatic N-Squared Charts. Hitchins proposes the use of a genetic algorithm to order and rearrange N-Squared Charts in a manner that reduces cognitive and perceptual complexity. The concept of N-Squared Chart configuration entropy is introduced and explained in Appendix A of the book. The example given in Hitchins' Figure 8.13 is addressed next using evolutionary computation and an abstract relation type formulation. The ART approach was again able to replicate the values presented in the published work. Figure 9 presents the initial and final ART marking spaces for the automated N-Squared Chart example, while Figure 10 presents the ART value space associated with this example.

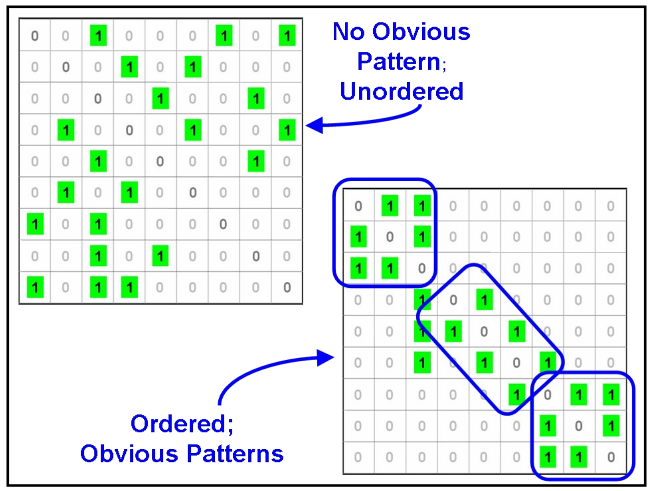

Figure 9 – *Initial and Final ART Marking Spaces for Automated N-Squared Chart Example.*

0	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7
2	1	0	1	2	3	4	5	6
3	2	1	0	1	2	3	4	5
4	3	2	1	0	1	2	3	4
5	4	3	2	1	0	1	2	3
6	5	4	3	2	1	0	1	2
7	6	5	4	3	2	1	0	1
8	7	6	5	4	3	2	1	0

Figure 10 – *ART Value Space for Automated N-Squared Chart Example.*

As shown in Figure 10, the Automated N-Squared Chart ART approach uses a symmetric value space to cluster the local system interconnections around the matrix diagonal with no bias to either the upper triangular matrix section or the lower triangular matrix section. The Design Structure Matrix approach gives more value to the connections in the lower triangular portion of the matrix as a mechanism to represent a "feed-forward" system sequence connection. Figure 11 presents the initial and final ART marking spaces for referenced Design Structure Matrix problem, while Figure 12 presents the ART value space associated with this specific problem.

Initial ART Marking Space

0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	1	0

Final ART Marking Space

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1	0	0	0	0
1	1	0	1	1	0	1	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0
0	0	0	0	1	0	1	0	1	0	0	0
0	0	1	1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	0	0	0	0	0	0

Figure 11 – *Initial and Final ART Marking Spaces for the Design Structure Matrix Example.*

0	43	44	45	46	47	48	49	50	51	52	53
21	0	42	43	44	45	46	47	48	49	50	51
20	19	0	41	42	43	44	45	46	47	48	49
19	18	17	0	40	41	42	43	44	45	46	47
18	17	16	15	0	39	40	41	42	43	44	45
17	16	15	14	13	0	38	39	40	41	42	43
16	15	14	13	12	11	0	37	38	39	40	41
15	14	13	12	11	10	9	0	36	37	37	39
14	13	12	11	10	9	8	7	0	35	36	37
13	12	11	10	9	8	7	6	5	0	34	35
12	11	10	9	8	7	6	5	4	3	0	33
11	10	9	8	7	6	5	4	3	2	1	0

Figure 12 – *ART Value Space for the Design Structure Matrix Example.*

8 Pragmatic Complexity Framework Outline

The ART technique was developed to provide a standard pattern for the representation of the many classical and current systems science and system engineering techniques that are available to assist the practicing systems professional in the identification, documentation, communication and modeling of complex systems. The pragmatic complexity framework is considered a higher level system abstraction than the ART technique. Almost all complex systems problems are associated with a large group of individuals from multiple technical, professional and social backgrounds. A key element in the evaluation of any complex system problem is the application of a language that communicates the system information at an abstraction level and application scope that is useful to the solution of the current system problem. Figure 13 provides an overview of the basic elements involved in the observation and identification of a system.

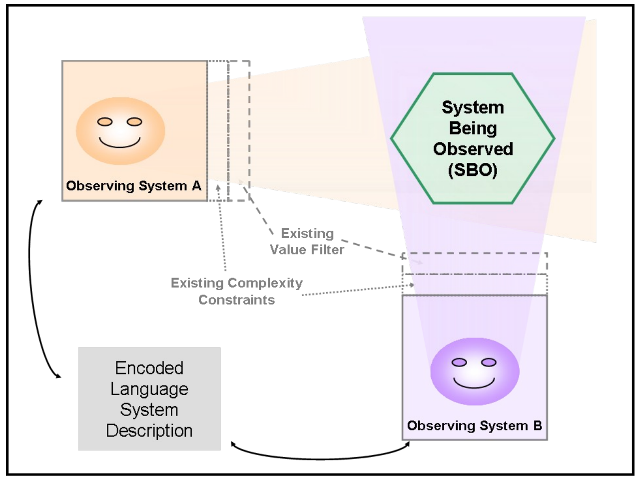

Figure 13 – *Pragmatic Complexity Framework Context.*

As shown in Figure 13, the pragmatic complexity framework is used to organize and arrange a set of ART artifacts into a specific set of encoded system patterns that represent both the global system relationships and the local system relationships in a manner that enhances technical understanding and the communication of system value. Figure 14 shows a typical system situation where each observer has limited information about the scope, form and purpose of the system being observed. Classical and/or current system engineering evaluation and analysis techniques are then applied using the pragmatic complexity framework to reduce the cognitive complexity associated with the specific system analysis task.

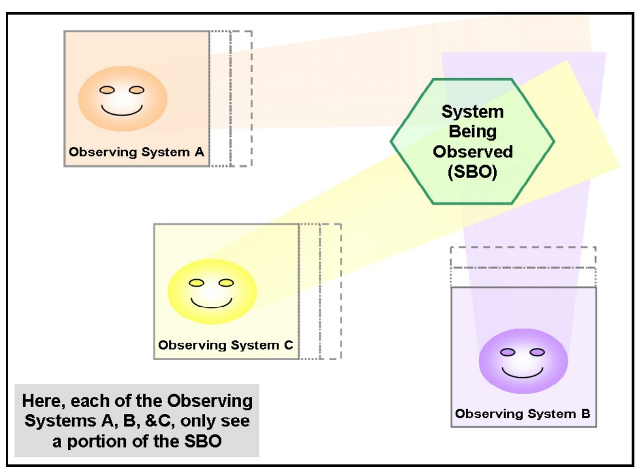

Figure 14 – *ART Pattern Application.*

The combination of evolutionary computation, standard systems analysis techniques, patterns and pattern languages in a manner that reduces cognitive complexity and increases the amount of information clearly communicated is the primary design goal of the pragmatic complexity framework. As discussed earlier in this paper, the same system can be represented by directed graphs that have different forms and semantics. This adaptable representation also applies to representing the system at different levels of abstraction and/or detail.

A key to this type of adaptive system representation is the separation of the representation of the global system structural relationship from the local system value determination. The ART construct cleanly separates these two aspects by creating and using a structural marking space as well a set of system value spaces. The two types of graphical links, structure and value provide the capability to identify a range of system structural configurations that conform to a given value set. Further, the global relationship that provides the logical basis for the construction of the ART marking space can be expressed at varying levels of aggregation and detail. The ability to vary the aggregation of the structural links in a primary mechanism is used in the reduction of cognitive complexity.

The general rule set to construct and maintain a pragmatic complexity framework includes the following steps:

1- Clearly establish a distinction between the pragmatic complexity framework development process and the framework artefact produced by the process.
2- Identify the system and/or systems of interest that are the subject of the pragmatic complexity framework.
3- Select the initial candidate set of ART constructs that will be used to implement the system pattern language associated with this specific application of the pragmatic complexity framework.
4- Organize the observing systems into functional groups determined by the level and scope of their observation capability.
5- Determine the set of global system relationships that will be used to evaluate the system being observed.
6- Determine the set of local value interconnections that will be used to evaluate any given system configuration and/or combinations of configurations.
7- Use the graphical, computational, prose and process elements of the ART components to effectively communicate information about the system being observed.
8- Iterate and refine the pragmatic complexity framework as necessary.
9- Iterate and refine the system pattern language as necessary.
10- Continue to reduce complexity in a pragmatic fashion.

9 Summary and Future Work

A pragmatic complexity framework has been introduced that reduces the cognitive complexity associated with large-scale systems engineering tasks. The framework facilitates the reduction of cognitive complexity in at least three basic ways: 1) developing a pattern language to support the consistent application of ART patterns; 2) developing ART forms that encode typical systems engineering techniques in clearly defined processes and patterns; and 3) supporting the application of computing resources to evaluate and analyze alternative system representations.

The clear separation of system structure from system value and semantics supports the detailed communication of both global and local system relationships. Further, the ART technique directly connects these relationships to mathematical relations that are the basis for the computation and analytical portions of the ART approach.

As mentioned earlier, some of the classical techniques were motivated by an attempt to reduce the computational complexity associated with computer problem solving. There is some indication that the ART technique can be used to directly reduce the computational complexity associated with the solution of complex system problems, this area will be explored in future work.

Future papers will publish detailed ART examples as well as example techniques that can be used to incorporate these ART forms into system patterns, system patterns languages and specific pragmatic complexity frameworks that reduce the complexity associated with system science, engineering and analysis tasks.

10 References

[1] Hall, A.D., Fagen, R.E. (1956), “Definition of a System”, *General Systems*, Vol. 1, pp. 18-28.
[2] Klir, G.J. (1969), *An Approach to General Systems Theory,* New York: Van Nostrand Reinhold Company.
[3] Warfield, J.N., Hill, J.D (1972), *A Unified Systems Engineering Concept, A Battelle Monograph, Number 1*, Columbus, OH: Battelle.
[4] .Steward, D.V. (1981), *Systems Analysis and Management Structure, Strategy, Design*, New York: A Petrocelli Book.
[5] Koth, W. (1995), Handbook of Semiotics, Bloomington, IN: Indiana University Press.
[6] Simpson, J.J., Simpson, M.J., (2003), “Systems and Objects”, *Proceedings, Thirteenth Annual International Symposium of INCOSE, "Engineering Tomorrow's World Today"*, Crystal City, VA: INCOSE.
[7] Warfield, J.N. (1974), *Structuring Complex Systems. Monograph No. 4,* Columbus, OH, Battelle Memorial Institute.

[8] Casti, J.L. (1989), *Alternate Realities: Mathematical Models of Nature and Man*, New York: John Wiley & Sons.
[9] Bar-Yam, Y. (1997), *Dynamics of Complex Systems*, Reading, MA: Perseus Books
[10] Warfield, J.N. (2003), *The Mathematics of Structure*, Boca Raton, FL: AJAR Publishing.
[11] Warfield, J.N., Cardenas, A.R. (1993, 2002), *A Handbook of Interactive Management*, Palm Harbor, FL: AJAR Publishing Company.
[12] Warfield, J.N. (1990), *A Science of Generic Design*, Ames: Iowa State University Press.
[13] Sage, A.P. (1977), *Methodology for Large Scale Systems*, New York: McGraw-Hill Book Company.
[14] Lano, R.J. (1979), *A Technique for Software and Systems Design*, Amsterdam: North Holland Publishing Company.
[15] Hitchins, D.K. (2003), *Advanced Systems Thinking, Engineering and Management*, London: Artech House, Inc.
[16] Simpson, J.J. (2004), "System Frameworks", Proceedings, *Second Annual Conference on Systems Engineering Research*, Los Angeles, CA: CSER2004.
[17] Simpson, J.J., Simpson, M.J. (2005), "System Integration Frameworks", *Proceedings, Fifteenth Annual International Symposium of INCOSE, "Systems Engineering: Bridging Industry, Government, and Academia"*, Rochester, NY: INCOSE.
[18] Simpson, J.J., Simpson, M.J. (2006a), "Formal Systems Concepts", *Proceedings, Fourth Annual Conference on Systems Engineering Research*, Los Angeles, CA: CSER2006.
[19] Simpson, J.J., Simpson, M.J. (2006b), "Systems Engineering (SE) Patterns and Pattern Language", *Proceedings, Fourth Annual Conference on Systems Engineering Research*, Los Angeles, CA: CSER2006.
[20] Simpson, J.J., Simpson, M.J. (2006c), "Foundational Systems Engineering (SE) Patterns for a SE Pattern Language", *Proceedings, Sixteenth Annual International Symposium of INCOSE, "Systems Engineering: Shining List on the Tough Issues "*, Orlando, FL: INCOSE.
[21] Simpson, J.J., Dagli, C., Miller, A (2007a), “System Evaluation and Description Using Abstract Relation Types (ART)”, *Proceedings of 1st Annual 2007 IEEE Systems Conference*, Honolulu, HI: IEEE.
[22] Simpson, J.J., Dagli, C., Miller, A (2007b), "Development and Application of Abstract Relation Types for Use in Systems and System-of-Systems Design and Evaluation", *Proceedings, Seventeenth Annual International Symposium of INCOSE, "Systems Engineering: Key to Intelligent Enterprises"*, San Diego, CA: INCOSE.
[23] Simpson, J.J., (2008a), "Evolutionary Computation in Systems Design and Evaluation", paper read to BTEC14, Boeing Technical Excellence Conference, February, Seattle, WA.
[24] Simpson, J.J., Dagli, C. (2008b), "Evolutionary Computation Applied to System Architecture Development", *Proceedings, Sixth Annual Conference on Systems Engineering Research*, Los Angeles, CA: CSER08.
[25] Simpson, J.J., Dagli, C. (2008c), "System of Systems Architecture Generation and Evaluation Using Evolutionary Algorithms", *Proceedings, 2008 IEEE 2nd International Systems Conference,* Montreal, Canada: IEEE.
[26] Simpson, J.J., Dagli, C. (2008d), "System of Systems Architecture Evaluation Using Evolutionary Computation", *Proceedings INCOSE*, Utrecht, Netherlands: INCOSE.
[27] Miller, G.A. (1956), “The Magical Number Seven, Plus or Minus Two: Some Limitations on Our Capacity for Processing Information”, Psychology Review 63(2), 81-97.
[28] Sharman, D.M., Yassine, A.A. (2004), "Characterizing Complex Product Architectures", *System Engineering Journal*, 7(1), 35-60.