Development and Application of Abstract Relation Types for Use in Systems and System-of-Systems Design and Evaluation

Joseph J Simpson	Dr. Cihan Dagli	Dr. Ann Miller
System Concepts	U of Missouri- Rolla	U of Missouri-Rolla
jjs-sbw@eskimo.com	dagli@umr.edu	milleran@umr.edu

Abstract. Abstract relation types (ART) are developed to represent, describe and establish a computational framework for a system. An abstract relation type is closely related to and builds upon two fundamental ideas. The first idea is the binary relation and structural modeling techniques developed by John N. Warfield. The second idea is the concept of abstract data types. These two ideas are combined to create an abstract relation type that provides a structured representation and computational method for systems and system components. The complete system description approach is based on six abstract relation types: context, concept, functions, requirements, architecture, and test (CCFRAT). When combined with digraphs and other graphical representations of the matrix form, ART provides a powerful tool for the communication of complex system interactions to large system design teams.

Introduction. System evaluation is a task that is becoming more difficult due to the increasing number of existing systems, interfaces and operations. At a basic level, a system can be defined in one of two general ways: a functional definition, or a "construction rule" definition. The functional definition of a system is "a constraint on variation." The construction rule definition of a system is "a relationship mapped over a set of objects." Abstract relation types assist in evaluation of systems that are defined or described in either of these two forms (Klir, 1969). A primary component of the ART construct, binary relations, have a long history of application in the systems science and engineering fields (Klir, 1991). Binary relations can be represented as matrices, directed graphs or lattices (Gratzer, 2003). The strong ties between the matrix form and graphical form of binary relations provide an excellent tool for system evaluation computation and communication. Structured group techniques that are based on binary relations and graph theory have been successfully used in the evaluation of large systems (Warfield, 2006).

ART are an important tool used to group contextual system relations with a set of visual and computational techniques to reason and communicate about systems. A well defined set of ART constructs will provide a common foundation for a systems engineering language and approach.

Abstract Relation Types. Two basic ideas are combined to create the ART. The first basic idea of structuring complex systems was developed by John N. Warfield (Warfield,

2003). This basic concept included the use of binary matrices, binary relationships and contextual relationships. A matrix **N** is represented by four sets; **N** = {*Is, It, Is x It, EIsxIt*}: where

Is is an ordered vertical index set
It is an ordered horizontal index set
Is x It is the set of all ordered pairs of *Is* and *It*
EIsxIt is the entry set (or content) of the matrix.

A binary relation, as defined by Warfield, is represented by a binary matrix A that defines a two-block partition {R; ~R}on *Is x It*, such that all ordered pairs in *Is x It* for which the entry is 1 are in the first block, and all other ordered pairs are in the second block. The first block is the binary relation R on *Is x It* and the second block is the complementary relation ~R on *Is x It*. A binary relation R on *Is x It* defines a binary matrix A indexed by *Is* and *It*. Some contextual relations developed by Warfield are: "is included in", "is antecedent to", "is subordinate to" and "is adjacent to." These relationships are mapped in a binary matrix and are used in the analysis of the structure of complex interactions and systems. (Warfield, 1974)

The second basic idea is the abstract data type. An abstract data type is used to organize the structure and specify the operations of a specific data type. The abstract nature of the data type ensures that the data type is independent of the underlying information types. (Rosen, 2000) Abstract relation types are not independent of the underlying relations. However, they use a predefined set of operations for each specific ART. An ART is composed of two primary components: a binary relation matrix and operations on the binary relation matrix. The operations are grouped into three basic types: those operations associated with the semantics of the relation, the binary matrix construction and evaluation, and the organization and display of the matrix information. ART extend the detail used in describing and defining the contextual relationship as developed by Warfield (1974).

Binary Relations. Warfield used relations that connect elements of the same set A, usually called relations on A. However binary relations can also apply to two sets, set A and set B. A binary relation between set A and set B links specific elements of set A with specific elements of set B. Binary relations have been applied in many areas in the practice of systems engineering. These areas include the practice of Interactive Management, N Squared Diagrams and Design Structure Matrices, which mostly use a set A on set A relation mapping.

Systems analysis, software evaluation and knowledge development areas use a binary, relation-based approach called Formal Concept Analysis. (Ganter, Wille) A formal context is constructed from two sets, A and B, and a relation I between A and B. The elements of set A are called objects and the elements of set B are called attributes of the context. The relation I is called the incidence relation of the context. The formal context structure is used to create a formal concept definition. These binary, relation-based structures have been used in software and knowledge engineering applications. Active research is continuing in the application of this technology in many engineering domains. The formal context binary relation shown in Figure 1 is used in the Context ART (XART). In this arrangement the context for any system is bounded by the systems that

are “visible” to the system of interest. As Figure 1 shows, the context for system 1A is the collection of systems, 1B, 1C and 1D.

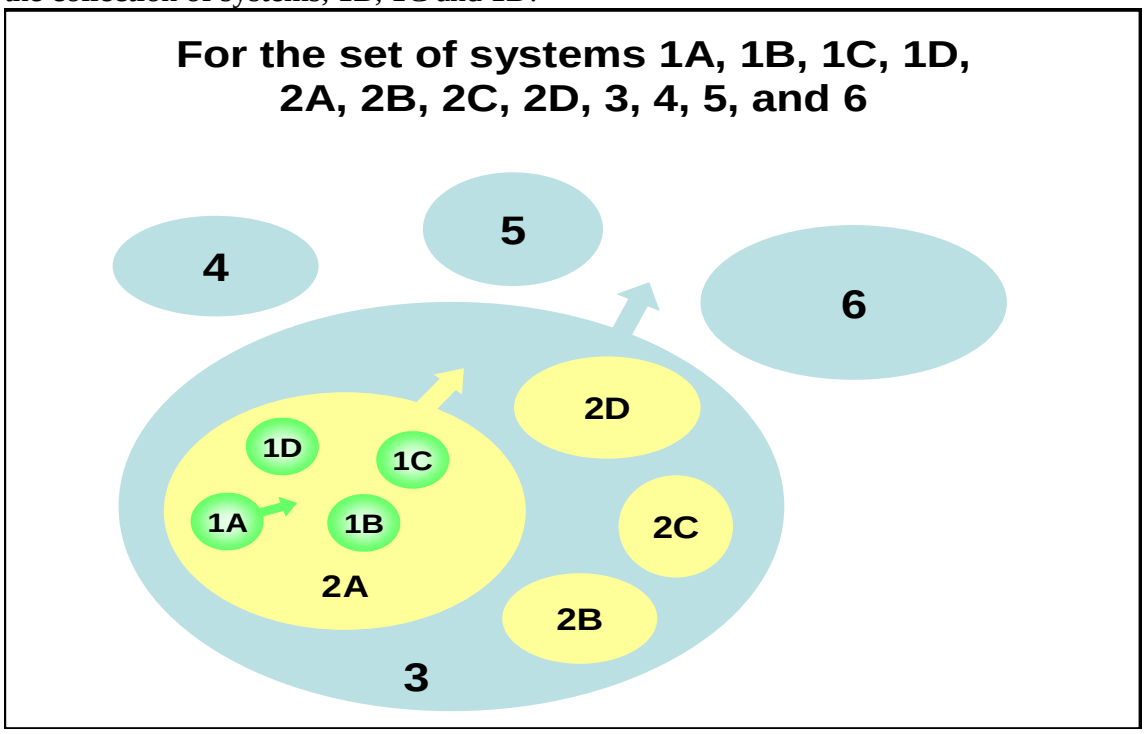

Figure 1 – System Context Space

The formal-context, binary relation for system 1A is shown in Figure 2. When the formal-context, binary relation is combined with a value set the basic XART is complete. The information encoded in the XART provides the foundation for many system-of-systems design and evaluation activities and decisions. Basic system connectivity and interoperability information can be read directly, or inferred, from the XART. For example, one might infer that system 1B has a wireless internet connection after reading the contextual information provided in Figure 2.

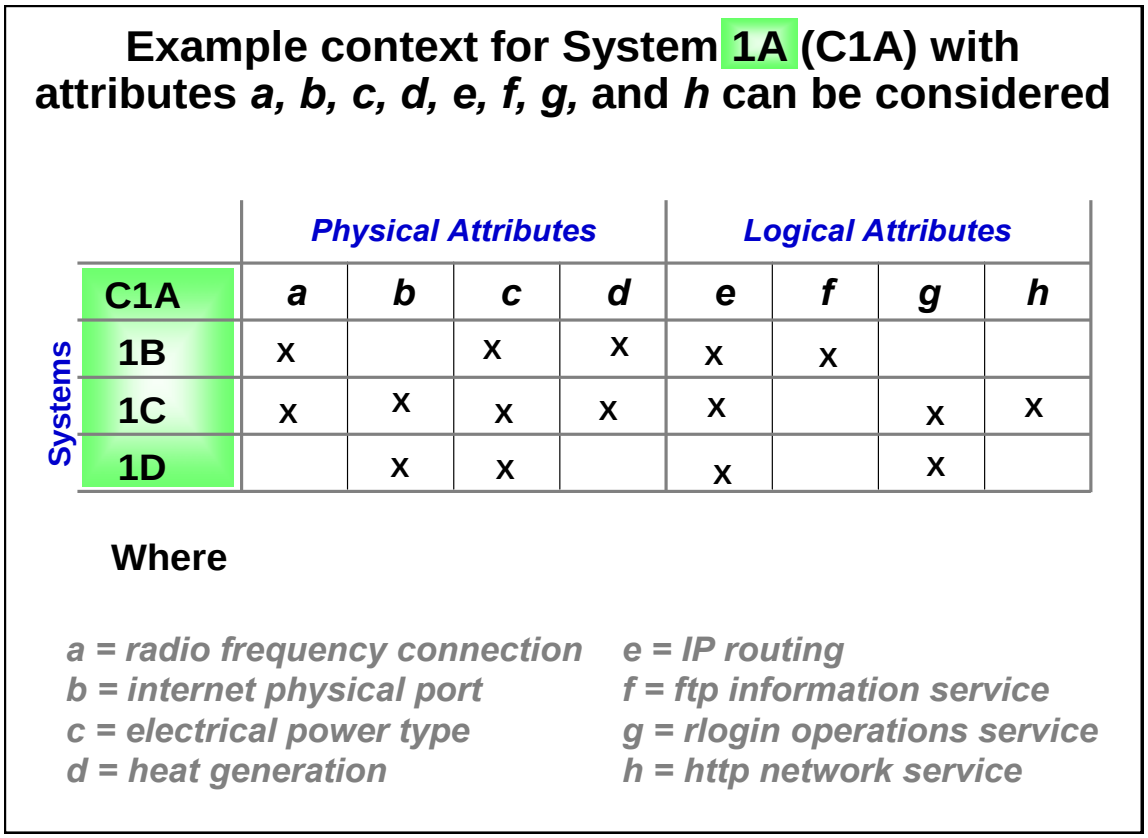

Figure 2 – Example Context for System 1A

Another application that uses the binary relation structure is the Union Rule Matrix Membership Array that is part of a fuzzy rule configuration technique developed by Combs. (Cox, 1998) The primary objective of this technique was the reduction of the number of rules required to model the complete system space. A binary matrix composed of two sets, individual attributes (set A) and a set of transitive scale values (set B). Combs provides an example of using both fuzzy values and discrete values. The individual attributes that are included in set A are age, health status, health history, job risk, and family risk. The scale values in set B can have different measurement units for each attribute in set A. However, each value scale must be transitive and have the same number of entries. The scales provided by Combs for each attribute are:

- Age: youthful, young, middle-aged, mature, old
- Health Status: excellent, good, average, below average, poor
- Health History: excellent, good, average, below average, poor
- Job Risk: minor, below average, average, above average, major
- Family Risk: minor, below average, average, above average, major

Further, these scale values in set B must be normalized in a manner that allows valid summation of the total set of measurements. Figure 3 provides an example of individual attributes for health insurance purposes (set A) and the relative amount of the insurance premium (set B). The numerical values in the lower matrix of Figure 3 represent fuzzy values, when the entries in each row must add up to one. The insurance premium rate is shown at the bottom of Figure 3 and calculated by selecting the column with the largest sum.

Binary Matrix: Set A X Set B

Set B

Set A	Youthful	Young	Middle-Aged	Mature	Old
HS	Excellent	Good	Average	Below Average	Poor
HH	Excellent	Good	Average	Below Average	Poor
JR	Minor	Below Average	Average	Above Average	Major
FR	Minor	Below Average	Average	Above Average	Major

A	0	0	0.25	0.50	0.25
HS	0	0.50	0.50	0	0
HH	0.25	0.50	0.25	0	0
JR	0.50	0.50	0	0	0
FR	0.25	0.50	0.25	0	0

RATE	Low	Mod-Low	Moderate	Mod-High	High

Figure 3 – Fuzzy Valued Matrix

Another application of a binary relation is the constraint matrix developed by Friedman as one of the four views of a mathematical model used in constraint theory. See Figure 4.

Set A (Relations) x Set B (Variables)

Variables

Relations	*X*	*x'*	*Z*	*z'*	*r*	*r'*	*S*	*s'*	*V*	*d*
1	●		●		●					
2	●	●	●	●		●				
3	●				●		●			
4			●		●		●			
5	●		●				●			
6	●	●	●	●				●		
7		●		●					●	
8		●		●						●

Where

1: $r = (X^2 + Z^2)^{1/2}$

2: $r' = (X'X + Z'Z)(X^2 + Z^2)^{1/2}$

3: $X = r \cos S$

4: $Z = r \sin S$

5: $S = \tan^{-1}(Z/X)$

6: $s' = (Z'X - X'Z)(X^2 + Z^2)^{-1}$

7: $r = (X'^2 + Z'^2)^{1/2}$

8: $Z = -X' \tan d$

Figure 4 – Constraint Matrix Meta Model

As used in constraint theory, a constraint matrix view displays the same information as the bipartite graph view and also acts as a meta-model that organizes information about the model's computational and structural properties. The constraint matrix view and form are used to facilitate the application of computer resources to the solution of constraint theory problems. (Friedman, 2005)

This collection of examples from the systems engineering literature indicates that binary matrices and binary relations have a long history of application and varied application in the practice of systems engineering.

Basic Set of System Abstract Relation Types. The basic set of system abstract relation types has six components. These six components are a relation types for context, concept, function, requirement, architecture, and test (CCFRAT). A specific component of the basic set may have more than one type of ART representation.

Two relation types, functional hierarchy (HART) and physical hierarchy (PART), are discussed in more detail in this paper than the other ART constructs. Each specific ART is identified by the relation type and the node type. For example, a HART would have functions as its nodes and "is included in" or "is composed of" as its binary relation. The methods associated with the HART support the operations of functional decomposition, abstraction level determination, and other semantically legal relation operations. A PART would have physical nodes (system, segment, subsystem….) and "is part of" or "is composed of" as its binary relation. The methods associated with the PART would support the operations of physical decomposition, abstraction level determination, and other semantically legal relation operations.

Functional Hierarchy Relation Type. The functional hierarchy relation type is based on an "is included in" and "is composed of" binary relation. Functional decomposition and functional flow block diagrams are classical system engineering methods that represent a system using this structural relation. An example of a functional decomposition is shown on the left hand side of Table 1.

Num	Function Name
0.0	Execute Task
1.0	Prepare for Task
1.1	Conduct Pre-task Briefing
1.2	Prepare Personnel for Task
1.3	Prepare Equipment for Task
2.0	Transit to Task Execution Area
2.1	Prepare for Transit
2.2	Load Unit on Transport
2.3	Move to Task Area
3.0	Perform Task
3.1	Evaluate Task Area
3.2	Execute Task Objectives
3.3	Evaluate Task Effectiveness
4.0	Return from Task Area
4.1	Transit to Initial Area
4.2	Evaluate Unit Status
4.3	Record Lessons Learned

Num	Component Name
0.0	Wagon
1.0	Wagon Body
1.1	Metal Floor
1.2	Metal Sides
1.3	Metal Fasteners
2.0	Wooden Sides
2.1	Wood Uprights
2.2	Wood Planks
2.3	Wood Fasteners
3.0	Front Wheel Assembly
3.1	Front Axel
3.2	Front Wheels
3.3	Front Handle Assembly
4.0	Rear Wheel Assembly
4.1	Rear Axel
4.2	Rear Wheels
4.3	Rear Axel Attachment Assembly

Table 1 – Examples of Functional versus Physical Hierarchies

The ‘Execute Task’ function is represented as the main function at level 0.0, or highest level of functional abstraction. At the next level of functional abstraction, four functions are presented to represent the higher level function: ‘Prepare for Task,’ ‘Transit to Task Execution Area,’ ‘Perform Task,’ and ‘Return from Task Area.’ Then each function is further elaborated by functions at the next lower level of functional abstraction. At the core of the HART is a binary matrix that represents the structure of this functional relation. Similar to abstract data types, ART have operations that are performed on the existing relation structure. These computational operations provide a mechanism to calculate, compute and reason about functional hierarchical relations.

Given the matrix structure of the HART, key characteristics and attributes of the structure can be precisely described, computed and communicated. Some of the characteristics and attributes are: level of abstraction, functional completeness at any level of abstraction, functional timing, and functional sequence. Figure 5 shows the translation of the execute task function into a binary “is included in” relation matrix. The binary relation matrix is read down the columns as “is included in.” As seen in the second column, a function is completely included in itself so the intersection of a function on the matrix diagonal contains a 1 that indicates the function is contained in itself. Column 2 also shows that the top level function, ‘Execute Task,’ is composed of all of the other functions listed in the matrix. For a binary relation matrix, the level of abstraction can be calculated from the number of cells in each row that contain the number 1. In the second row we see that there is only one cell with the number 1 in the row, so this is the highest

level function. The third, fourth, fifth and sixth rows all contain two cells that contain the number 1; these are second level functions. The rest of the rows all contain three cells that contain the number 1 indicating that these are level three functions, the lowest level in this function set.

	0.0	**1.0**	**2.0**	**3.0**	**4.0**	**1.1**	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**3.3**	**4.1**	**4.2**	**4.3**
0.0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	**1**	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0
1.1	**1**	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0
1.2	**1**	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0
1.3	**1**	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
2.1	**1**	0	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
2.2	**1**	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0
2.3	**1**	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
3.1	**1**	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0
3.2	**1**	0	0	**1**	0	0	0	0	0	0	0	0	**1**	0	0	0	0
3.3	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0	0
4.1	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0
4.2	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**	0
4.3	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**

Figure 5 – Binary “Is Included In” Matrix

Once the binary matrix structure has been established for the HART, other operations can be performed on the matrix. For example, the level two functions shown in Figure 5 are sequential in nature. Therefore, sequential logic and constraints can be applied and evaluated by HART methods. The top level function may have specific time, energy and/or other constraints. Because the second level functions are sequential, the time constraint can be allocated across these level-two functions. The identification of a sequential function context relation allows the computation of total time for the top level function. The time for each lower level function may be added for a total time, or the total time required by the highest level function may be budgeted as a time constraint to each lower level function.

The HART provides the foundation for the development of “sequential function” and “concurrent function” operators and binary relation matrix calculations. Using these techniques, large complex functional structures can be developed, analyzed and communicated across large distributed system development teams in a precise manner.

Physical Hierarchy Relation Type. The physical hierarchy relation type is based on an “is part of” or “is composed of” binary relation. Physical system decomposition is an example of this type of systems analysis activity. An example of a physical decomposition for a wagon is shown on the right side of Table 1. The ‘Wagon’ component represents the highest level of physical composition. The hierarchical structure of the physical decomposition was selected to match the hierarchical structure of the functional decomposition to highlight the fact that even though the binary structure is the same, the operations and semantics of the PART have a different meaning than the operations and semantics associated with the HART. Physical properties and attributes

are important aspects used in PART methods and computations. Weight is an example of a physical property that can be allocated, from the top-down, or computed, from the bottom-up in the PART. As with the HART, each level of physical decomposition can be clearly computed and communicated using PART methods. Using the "is a part of" and "is included in" binary relation, the wagon physical decomposition would look exactly like the binary matrix in Figure 5. The second column, of Figure 5, indicates that all of the physical components are included in the wagon. In a manner similar to the HART methods, each row would be evaluated to calculate the proper level of physical decomposition. Even though column 2 of Figure 5 indicates that all of the physical components are included in the wagon, the physical level of decomposition must be taken in to account when PART methods are calculating a complete system. The semantics associated with each relation and ART will cause the methods associated with the ART to change depending on the ART type.

An example of using the PART to represent and evaluate system weight is shown in Figure 6. In this case the numbers in the cells represent the allocated weight for each physical component. The total wagon weight, 26 units, is located in cell (2, 2). Summing the weight of all level two components; rows 3, 4, 5, and 6, gives the expected weight of the total system. Summing the weight of all of the level three components; rows 7 through 18 again gives the expected weight for the complete system.

	0.0	**1.0**	**2.0**	**3.0**	**4.0**	**1.1**	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**3.3**	**4.1**	**4.2**	**4.3**
0.0	**26**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	**10**	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	**6**	0	**6**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	**6**	0	0	**6**	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	**4**	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0	0
1.1	**4**	**4**	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0
1.2	**4**	**4**	0	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0
1.3	**2**	**2**	0	0	0	0	0	**2**	0	0	0	0	0	0	0	0	0
2.1	**2**	0	**2**	0	0	0	0	0	**2**	0	0	0	0	0	0	0	0
2.2	**3**	0	**3**	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0
2.3	**1**	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
3.1	**2**	0	0	**2**	0	0	0	0	0	0	0	**2**	0	0	0	0	0
3.2	**2**	0	0	**2**	0	0	0	0	0	0	0	0	**2**	0	0	0	0
3.3	**2**	0	0	**2**	0	0	0	0	0	0	0	0	0	**2**	0	0	0
4.1	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0
4.2	**2**	0	0	0	**2**	0	0	0	0	0	0	0	0	0	0	**2**	0
4.3	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**

Figure 6 – Physical Component Weight Allocation

Genetic Algorithms and Abstract Relation Type Utilization in System Evaluation. Once the functional abstraction levels have been clearly defined using a HART, the functional interactions at each level may be evaluated using an "interfaces with" and "does not interface with" binary matrix. These types of binary relationship matrices are more commonly called N Squared diagrams. Genetic algorithms are an excellent tool to assist in the search for optimal solutions in these types of structured problems (Coley, 2005). Given a specific definition of the interface type and interface values, genetic algorithms may be used to find the interface arrangement that has the best or highest

value configuration (Hitchins, 2003). Two examples of the application of genetic algorithms to system evaluation are shown in Figure 7. The top two matrix representations, A1 and A2, in Figure 7 show the application of a genetic algorithm to the ordering of the level three functions from Figure 5. Figure 7, matrix A1, shows that the level three functions from Figure 5 are not strictly sequential like the level two functions in Figure 5. The "disordered" interfaces in the first matrix, A1, are ordered using a genetic algorithm and selection metric proposed by Hitchins and the resulting matrix is marked as A2 (Hitchins, 2003). The second set of matrices, A3 and A4, were taken from a design structure matrix (DSM) clustering example (Sharman and Yassine, 2004). The same type of genetic algorithm and selection matrix was used to evaluate the DSM example.

The binary matrix representation of the "interfaces with" relation provides a number of benefits for genetic algorithm application. The binary matrix representation of the chromosome "building blocks" are completely specified with fixed order in one dimension, while the other dimension provides the area of design combination. The value of the selection fitness function is provided as a function of another sequentially ordered matrix (Knjazew, 2002).

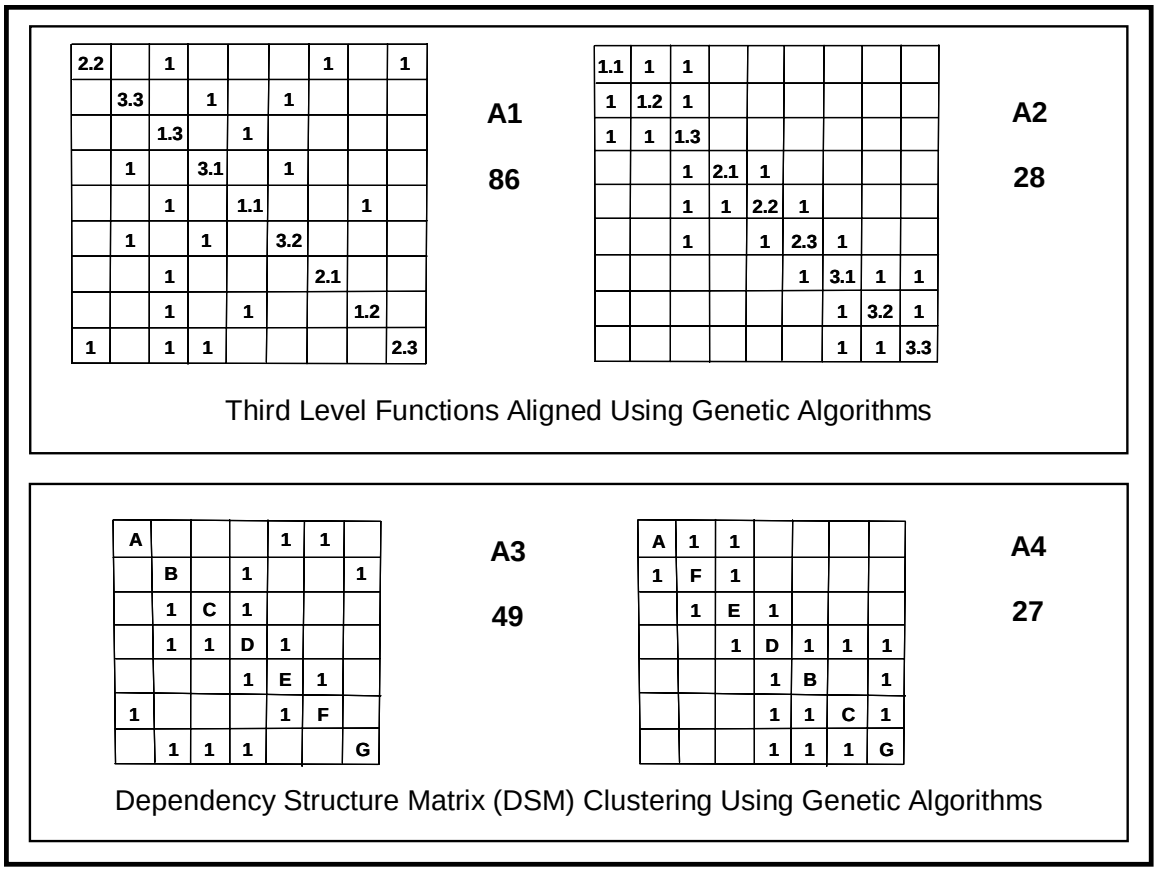

Figure 7– Application of Genetic Algorithms to System Evaluation

Summary and Conclusions. This paper has introduced the idea of an abstract relation type, which encapsulates a binary relation and matrix with a set of computational operations. The twin foundations of natural language context relationships and the binary

matrix form provide a solid basis upon which intelligent system computation can be based. Two of the six basic system ART constructs were discussed. These two ART cover most of the primary system design drivers. The functional analysis activity that creates the HART is usually one of the first steps in the system design process. The physical system design and structure that is represented by the PART is developed prior to system production. Creating the best value physical system that provides the required system functions is the primary goal of systems engineering. Incorporating an ART computational framework in the system production process will provide the foundation for the application of computational intelligence techniques as well as the clear communication of primary system design parameters. More research and development needs to be applied to the basic set of ART to create a robust system evaluation and design technique. There are many strong similarities between the genetic algorithm approach developed by Derek Hitchins (2003) and the ordered messy genetic algorithm developed by Dimitri Knjazew (2002). More research is required to evaluate the value provided by these two approaches to the description, design and evaluation of systems using genetic algorithms.

References.

Coley, David A., *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, New Jersey, 2005

Cox, Earl, *The Fuzzy Systems Handbook, Second Edition*, AP Professional, New York, 1998.

Friedman, George, *Constraint Theory, Multidimensional Mathematical Model Management*, Springer Science, New York, 2005

Ganter, B., Wille, R., *Formal Concept Analysis, Mathematical Foundations*. Springer-Verlag, New York, 1999.

Gratzer, George, *General Lattice Theory*, Birkhauser Verlag, Boston, 2003

Hitchins, Derek K., *Advanced Systems Thinking, Engineering, and Management*, Artech House, London, England, 2003.

Klir, George J., *Facets of Systems Science*, Plenum Press, New York, 1991

Klir, George J., *An Approach to General Systems Theory*, Van Nostrand Reinhold Company, New York, 1969

Knjazew, Dimitri, *OmeGA, A Competent Genetic Algorithm for Solving Permutation and Scheduling Problems*, Kluwer Academic Publishers, Boston, MA, 2002

Rosen, Kenneth H., *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, New York, 2000

Simpson, J. Dagli C., Miller A., Grasman S., "A Generic, Adaptive Systems Engineering Information Model", *Proceedings of the 15th Annual International Symposium of the International Council on Systems Engineering* (Rochester, NY, July, 2005)

Warfield, John N. *An Introduction to Systems Science*, World Scientific, New Jersey, 2006.

Warfield, John N. *The Mathematics of Structure*, AJAR Publishing, Palm Harbor, FL, 2003.

Warfield, John N. *Structuring Complex Systems*, Battle Monograph No. 4, Battelle Memorial Institute, Columbus , OH, 1974.

Biography.

Joseph J. Simpson's interests are centered in the area of complex systems including system description, design, control and management. Joseph has professional experience in several domain areas including environmental restoration, commercial aerospace and information systems In the aerospace domain, Joseph has participated in a number of system development activities including; satellite based IP network design and deployment, real-time synchronous computing network test and evaluation, as well as future combat systems communications network design. Joseph Simpson has a BSCE and MSCE from the University of Washington, an MSSE from the University of Missouri-Rolla, is a member of INCOSE, IEEE, and ACM. Currently Joseph is enrolled in a system engineering doctorate program at the University of Missouri-Rolla.

Dr. Cihan H Dagli is a Professor of Engineering Management and Systems Engineering and director of the System Engineering graduate program at the University of Missouri-Rolla. He received BS and MS degrees in Industrial Engineering from Middle East Technical University and a Ph.D. from the School of Manufacturing and Mechanical Engineering at the University of Birmingham, United Kingdom, where from 1976 to 1979 he was a British Council Fellow. His research interests are in the areas of Systems Architecting, Systems Engineering, and Smart Engineering Systems Design through the use of Artificial Neural Networks, Fuzzy Logic, and Evolutionary Programming. He is the founder of the Artificial Neural Networks in Engineering (ANNIE) conference being held in St. Louis, Missouri since 1991. He provided the conduit to the dissemination of neural networks applications in engineering and decision making through these conferences for the last fourteen years. He is the Area editor for Intelligent Systems of the International Journal of General Systems, published by Taylor and Francis, and Informa Inc.

Dr. Ann Miller is the Cynthia Tang Missouri Distinguished Professor of Computer Engineering at the University of Missouri – Rolla. Previously, she was the Deputy Assistant Secretary of the Navy for Command, Control, Communications, Computing, Intelligence, Electronic Warfare, and Space for the U. S. Department of the Navy. For a portion of that time, she had additional responsibilities as Department of the Navy Chief Information Officer (CIO). She also served as Director for Information Technologies, Department of Defense Research and Engineering. Prior to that, Dr. Miller served for over 12 years with Motorola, Inc. where she held a variety of technical and managerial positions. She holds one U. S. patent in satellite communications, has co-authored three books on the programming language Pascal, and is the author of more than five dozen journal articles and monographs. Dr. Miller chairs the NATO Information Systems Technology Panel and is a Senior Member of IEEE. Dr. Miller's research areas include reliability and security of computer-based systems, with an emphasis on networked large-scale systems.