15th ICCRTS

"The Evolution of C2"

Title of Paper: A Foundation for Evolving Command and Control Activities

Topic 1: C2 Concepts, Theory, and Policy
Topic 9: C2 Architectures and Technologies
Topic 3: Information Sharing and Collaboration Processes and Behaviors

Norman K. Ma, Timothy Aiken

Point of Contact: Norman K. Ma

MITRE
202 Burlington Road
Bedford, MA 01730-1420

(979) 571-7777
nma@mitre.org

# A Foundation for Evolving Command and Control Activities


## Abstract

This paper describes a theory of constant foundation for evolving C2 activities through time and space. The theory is conducive to C2 domain analysis and engineering implementation. Based on an established scientific problem-solving paradigm in complex space, the Common Agility Space (CAS) representation, to enable the description, measurement, and analysis of C2 activities. CAS codifies two C2 perspectives: 1. (physical, information, cognitive, social) and 2. (tactical, operational, strategic). In addition to providing a unifying big picture for strategic activities, the CAS representation is instantiatable using existing scientific and engineering tools to assist composable edge tactical activities. Structural momentum of an organization and abundance of unfiltered information are two overhead factors that can consume inordinate resources to support decision making. The CAS theory is able to integrate the innate organization processes, environmental inputs, and organizational intentions to align resources to achieve desired effects. This paper first gives an introduction to the CAS foundational problem solving paradigm, explains how CAS is a flexible and unifying representation of C2 activities, describes situation assessment and Effect Based Operation (EBO) using the CAS context, gives a scenario using CAS, and finally briefly lists some possible extension of CAS in both the theoretic and engineering directions.

A Foundation for Evolving Command and Control Activities

## Overview

A single abstracted approach by a nation's defense system is insufficient to generate C2 (command and control) responses that effectively address destabilizing scenarios such as extreme weather [G1], natural and intentional biological diseases [D1], socio-organizational discords [C1], distributed asymmetric terrorism [E1], and recently, internal and external economic instability [B1]. On the other hand, an un-coordinated distributed approach is also insufficient to generate consistent commands. This writing proposes a common construct named Common Agility Space (CAS) that spans the physical, information, cognitive, and societal domains [S2] by providing a common perspective between C2 tools, processes, requirements, and users as a step in increasing C2 effectiveness and connect evolving C2 perspective, methodologies, and doctrines. Specifically, CAS is a construct [N1] that bridges the environment and C2 concepts:

- Situation Awareness [E2] identifies states in a CAS model.
- Effects Based Operation [S3] selects states in the CAS as targeted effects.
- Agility [A2] is a characteristic of the paths amongst the CAS states.

Figure 1 shows a relationship with focus on some of the major themes of command and control Agility, Situation Awareness and Effects-based Operation that are united by their underlying problem space. The figure also includes some C2 research sub-areas including processes, tools, requirement, and various C2 roles summarized as users. We examine the implicit C2 problem space induced by the modeling of environmental objects for the purpose of generating responses using information technology. However, issues concerning the engineering of a common problem space have been left for future investigation.

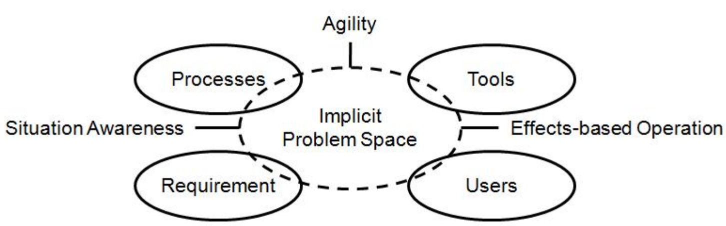


Figure 1, Focus on a Common Problem Space for C2 Activities

C2 activities seek to transition a system to a stable state in an environment that contains multiple forces [A1]. These forces permeate the environment, when viewed through the battle-space perspective; generate activities in multiple battle-space domains which require response and consumption of C2 resources to stabilize a system. These forces may be known or unknown, natural or intentional, sporadic or pandemic. Thus an empowered C2 organization [B2] needs to assure its strategic commands are encompassing to meet goals, its operational controls are responsive, and the ramifications of its tactical activities move the system towards desired effects. The CAS construct is able to represent all three levels of C2 activities. A path of stabilizing a system involves using command and control methodologies to negotiate in the physical, informational, cognitive, and societal domains. Network centric technologies are a means to improve situation awareness in order to reach desired effects through efficient operations; all tactical activities are carried out in an innately complex problem space where the approach to describe the problem space is significant.

## Agility of Problem Space

Agility is defined as an ability of an organization "to adapt, to learn, and to change to meet the threats that they face" [A2]. Interestingly, that description is similar to the description of *intelligence* by many artificial intelligence researchers. Atkinson further describes a system as agile if it provides users with more options. Agility has an implied meaning of flexibility and speed. When used in the context of C2 processes and tools, it is a reasonable descriptive word that describes how quickly tools assist processes in reaching an intended effect. That is, a C2 tool should support its users in reaching targeted effect quickly in wide possible situations, and similarly a process is more agile if its execution enables a system to reach an effect quicker than other processes. The Common Agility Space can be thought of as a common construct for C2 tools. A realistic theory and accurate situation assessment (such as assembling multiple perspectives to form a higher understanding of the situation) can generate actions that affect the environment towards a goal.

## Problem Space Development

The Common Agility Space is a state-space based problem solving representation based on graph theory [Z1]. A CAS model is composed of vertices that represent the possible *states* of the system that is being modeled and *edges* that connect the states. Development of the model is a continuous process. Each state represents a possible collection of values of the problem space variables. Using a dining scenario as example; we start by modeling a group dinner scenario with two variables, the progress of a dinner and location of the event. Figure 2 also includes possible values of dinner status and where the dinner location:

*Dinner Status* = {Complete, In Progress, Not Started},
*Dinner Location* = {At Home, Eat Out}.

Figure 2, Two Variables Describing a Dinner Scenario

There are 6 possible states from the multiplication of the 3 possible values [S1] of *Dinner Status* and 2 values of *Dinner Location*. The present state of the dining scenario is one of these 6 possibilities induced from the 2 variables. Edges between the states describe how the system state can change via events or actions that are not shown here. Each state should have at least an action either entering or exiting that state. At this point, we take a closer examination of how complexity [M1][S2] increases in this particular problem space. To describe the type of food, an additional variable can be added to the model as in Figure 3 below,

*Cuisine Type* = {American, Chinese, Cuban, German,
Indian, Italian, Japanese, Mexican, Thai}.

Figure 3, Two Additional Variables Describing the Same Scenario

then the size of the problem space would increase significantly from 6 to 54. This is from multiplying 6 by the 8 added possibilities from the *Cuisine Type* variable. A visualization of the increase in the problem space size is shown in Figure 4. This large increase in the size of the problem space due to just one additional variable characterize the behavior of C2 and other real-world problem spaces, and more importantly, causes degradation of user and tool capabilities that support C2 activities.  If a C2 activity takes 5 seconds to examine each state, then the total time to exam the problem space would increase more than an order of magnitude from 30 seconds to 270 seconds. When additional variables are used to create an even more realistic model, the increase would be much more dramatic.

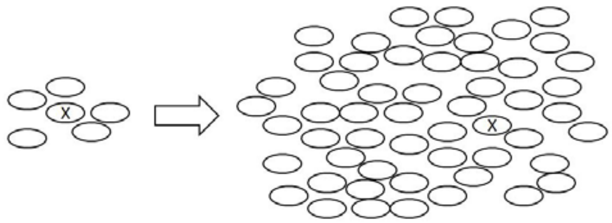

Figure 4, Disproportional Problem Space Increase after Adding One More Variable

By adding one more variable to the problem space, we investigate the characteristic of the growth of the problem space and proceed with a comment on the significance of the *approaches* in building a CAS model. The next variable, *Planner*, describes the person who is responsible for planning the dinner to complete the dinner scenario model:

*Planner* = {Shane, Rose, Ahmed, Kendall}

A possible state of a model is where each variable that describes part of the model has a specific value, as in Figure 5 below, where a group of people are having dinner at a Mexican restaurant that was selected by Kendall.

{(Dinner Status: In Progress) (Dinner Location: Eat Out)
(Food Type: Mexican) (Planner: Kendall)}.

Figure 5, Example of a Specific State

The size of the problem space is now $216 = 3 \cdot 2 \cdot 9 \cdot 4$ and the time would increase by an order of magnitude from 270 to 1080 seconds to examine each state of the newly evolved problem space. A notable point from this example is the problem space grows quickly when building representation of scenario. Processes or systems that function effectively at one time can quickly become ineffective after problem space change. A specific result of the change can manifest itself as a large increase in the possibilities that need to be considered when incremental considerations are added to the picture. In addition to the increase in size, it is possible that the characteristic of the space can change, such as the distribution of solutions and the available paths to those solutions, which is visualized in Figure 6.

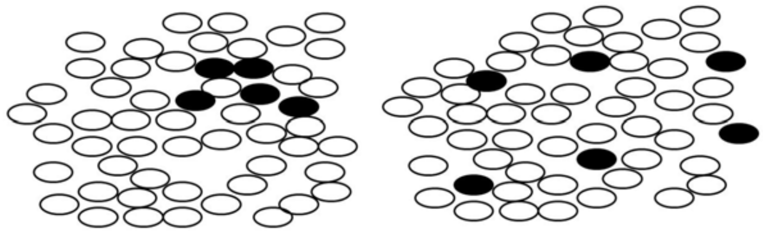

Figure 6, Change in the distribution of solutions in a problem space

The size of the problem space is the same for both problem spaces in Figure 6, but the goal states are distributed differently. The goal state are clustered together at the left problem space while dispersed widely in the right problem space. The implication is that a C2 tools that searches for these goal states can perform differently on similar sized problems. For example, locating a person in a city at Canada versus in a similar sized city at Nigeria. In the above dinner scenario example the size of the problem space is easily quantifiable, higher fidelity models can contain uncountable states. However, existing algorithms, heuristics, and large body of research exists for exploring large problem spaces. As a problem space is being iteratively constructed and information technology applied to the evolving construct, we present evidence that the *approach* to the construction of the problem space is important. Take the above dining example of 4 variables with a total of 216 possible states. There are $4 \cdot 3 \cdot 2 \cdot 1 = 24$ approaches in constructing the problems space. For example, we could first decide what type of food the group wants then appoint a person to choose a restaurant, or the group could choose either dine-in or take-out first then decide on other questions. An enumeration of all possible approaches is represented in Figure 7. Each row begins with the approach number and followed by a sequence of dinner scenario variables being added to the problem space. For example, row 1 starts with the number 1 and followed by 1, 2, 3, and 4. That means this particular problem space is constructed by first adding the *Dinner Location* variable, followed by the *Dinner Status* variable, and then *Planner*, *Food Type* variables:

| | First | Second | Third | Fourth | | First | Second | Third | Fourth |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 13 | 3 | 1 | 2 | 4 |
| 2 | 1 | 2 | 4 | 3 | 14 | 3 | 1 | 4 | 2 |
| 3 | 1 | 3 | 2 | 4 | 15 | 3 | 2 | 1 | 4 |
| 4 | 1 | 3 | 4 | 2 | 16 | 3 | 2 | 4 | 1 |
| 5 | 1 | 4 | 2 | 3 | 17 | 3 | 4 | 1 | 2 |
| 6 | 1 | 4 | 3 | 2 | 18 | 3 | 4 | 2 | 1 |
| 7 | 2 | 1 | 3 | 4 | 19 | 4 | 1 | 2 | 3 |
| 8 | 2 | 1 | 4 | 3 | 20 | 4 | 1 | 3 | 2 |
| 9 | 2 | 3 | 1 | 4 | 21 | 4 | 2 | 1 | 3 |
| 10 | 2 | 3 | 4 | 1 | 22 | 4 | 2 | 3 | 1 |
| 11 | 2 | 4 | 1 | 3 | 23 | 4 | 3 | 1 | 2 |
| 12 | 2 | 4 | 3 | 1 | 24 | 4 | 3 | 2 | 1 |

Figure 7, Approaches in Constructing a 4-Variable Problem Space

The ramification of this enumeration is the rate at which the problem spaces unfold, as shown below in Figure 8. For example, row 1 contains the number 1 then 2, 6, 24, and 216. The first number is just a label. The numbers 1, 6, 24, 216 indicates the size of the problem space after variables *Dinner Location*, *Dinner Status*, *Planner*, and *Food Type* are added respectively.

| | First | Second | Third | Fourth | | First | Second | Third | Fourth |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 6 | 24 | 216 | 13 | 4 | 8 | 24 | 216 |
| 2 | 2 | 6 | 54 | 216 | 14 | 4 | 8 | 72 | 216 |
| 3 | 2 | 8 | 24 | 216 | 15 | 4 | 12 | 24 | 216 |
| 4 | 2 | 8 | 72 | 216 | 16 | 4 | 12 | 108 | 216 |
| 5 | 2 | 18 | 54 | 216 | 17 | 4 | 36 | 72 | 216 |
| 6 | 2 | 18 | 72 | 216 | 18 | 4 | 36 | 108 | 216 |
| 7 | 3 | 6 | 24 | 216 | 19 | 9 | 18 | 54 | 216 |
| 8 | 3 | 6 | 54 | 216 | 20 | 9 | 18 | 72 | 216 |
| 9 | 3 | 12 | 24 | 216 | 21 | 9 | 27 | 54 | 216 |
| 10 | 3 | 12 | 108 | 216 | 22 | 9 | 27 | 108 | 216 |
| 11 | 3 | 27 | 54 | 216 | 23 | 9 | 36 | 72 | 216 |
| 12 | 3 | 27 | 108 | 216 | 24 | 9 | 36 | 108 | 216 |

Figure 8, Problem Space Growth based on Approach

While the final problem space size, 216, is the same for all approaches, the size of the problem space increases at different rates. This can have a significant impact on decision making. Figure 9 is a graph of the rate of growth from Figure 8.

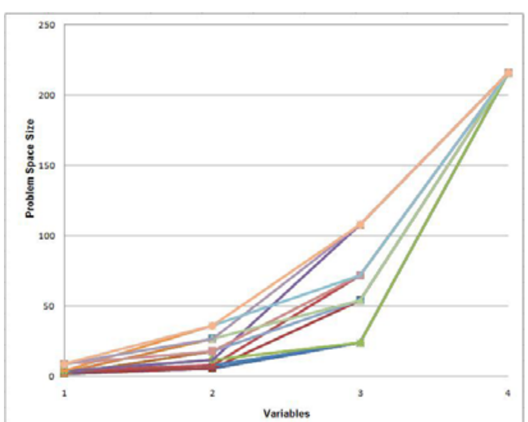


Figure 9, Approach Effects on Problem Space Growth

The vertical axis indicates the size of the problem space and the model variables are added from left to right along the horizontal axis. While the ending size is the same, there are significant differences in the growth of the problem space. The graph in Figure 10 below shows two of the approaches fitted with power curves

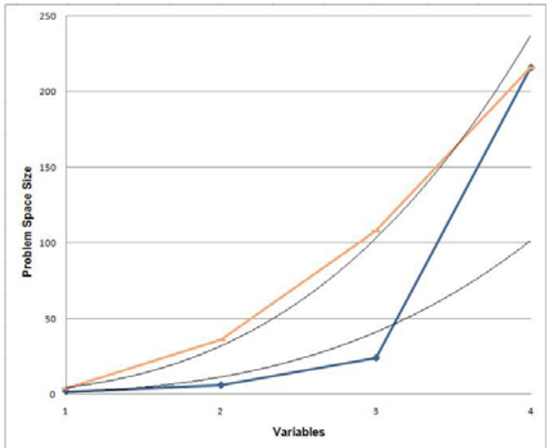


Figure 10, Use Power Curve Fit to Describe Problem Space Approach

We use standard fitted power curve to describe problem space growth as more variables are added to a model. A couple of reasons of adding more variables are to increase the fidelity of the model and/or accentuate a perspective. A simple power curve is composed of a parameter, a variable, and an exponent

$$parameter \cdot variable^{exponent}$$

We focus on the exponent of the fitted power curves that describe the different approaches (which variables are being added to the model) in building a problem space because the exponent characterizes the expansion of problem space. An exponent that is less than or equals to 1.0 indicates that particular problem space will expand at or below the rate in which variables are being added to the model. This is demonstrated when a user desires higher fidelity from a model (e.g. to answer more questions) and variables are added to the model to reach higher fidelity. The resulting problem space will increase in size relatively slowly and thus consume less resources (computation and cognitive) to derive a solution. On the other hand, if a problem space is characterized with an exponent greater than 1.0, then the amount of additional resources (computing or brain power) needed can be disproportionally large due to a large increase in search space, and thus increasing the likelihood of ungraceful tool degradation and overwhelming an individual's cognitive ability. What is interesting here is that the approach being used can be significant when expanding a problem space. To have a visual understanding of problem space growth, Figure 11 plots the power curve exponents of the 24 approaches based on standard fitting of power curve

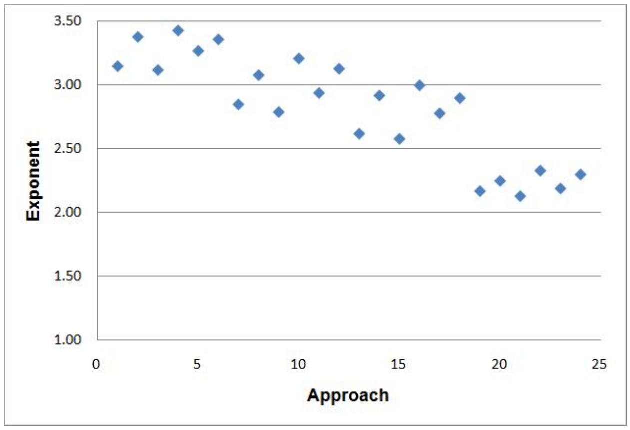


Figure 11, Problem Space Growth Rates due to Problem Space Construction Approaches

The range of exponents due to different approaches is in 2.13 to 3.43. This indicates that even this seemingly simple dinner problem induces a complex problem space, therefore consumes disproportionally larger processing resource as the model is being constructed. That said, even within a complex problem domain, the *approach* towards a problem can make a difference. Figure 12 shows the growth of the problem space of the best and the worst approaches of the dinner problem description

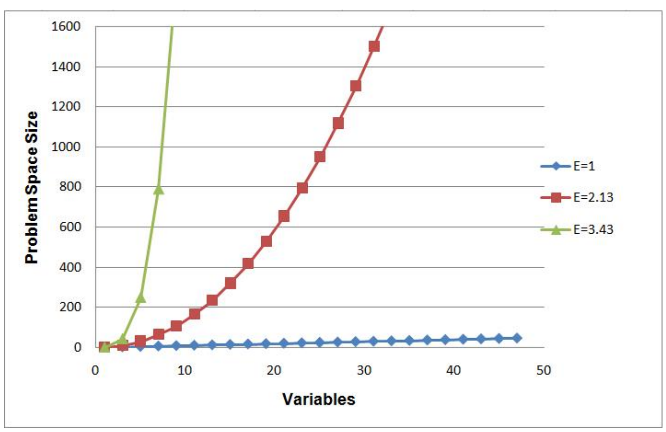


Figure 12, Range of Problem Space Growth Rates due to Approach

The problem space grows quickly when exponent is 3.43, as shown by the green-triangle curve. The size appears to be intractable after 10 variables have been incorporated into the model. On the other hand, more than twice the number of variables can be considered when building the problem space using a different approach (exponent is 2.13), as shown by the red-square curve. What does this mean practically? An implication is that existing tools and process that support C2 activities would become ineffective as a command and control system is challenged to operate in more destabilizing scenarios. In addition, the method of approach in analysis has an impact on the complexity and thus effectiveness of C2 activities. Moreover, these tools and processes can break-down unexpectedly if the problem representation is being constructed in an intractable manner. Many C2 problem spaces are instantiated and grow organically in both user's cognitive domain and tool's information domain. It is unlikely that an all encompassing investigations has taken place in identifying the best approach in constructing an explicit problem space.  Further study is needed to answer these questions and improve existing command and control capabilities.

## Segmenting Domain Space

A method of improve the intractable growth of problem space described above is to segment the problem space. Limiting C2 activities within a segmented problem space and processing each segmented problem in a sequential or parallel manner can stave off the failure-point of C2 tools and processes. Moreover, we demonstrate that segmenting the problem space into physical, information, and cognitive domains also is reasonable for the purpose of building an encompassing representation of reality. We begin with a simple model; Figure 13 below is part of a model from a suicide bombing scenario:


Figure 13: Vertex and Edge of a Physical Domain Model

This model shows that the explosive vest is located on the table top and if an action was taken where the potential bomber signs the pledge, the location of the vest would still be on the table, but the situation has changed. This state contains a single variable *Vest Location* with a value of *Table Top*, other possible values of the variable are {*Lat_Long, Under Construction, Being Worn, Unknown*}, thus this model contains at least four states. Higher fidelity models would consider more than one variable; the size of the model space is the mathematic product (multiplication) of the possible values of the variables. While the bomber signing a suicide pledge does not change the physical model's state, there is definitely an increase of danger in the environment. The information domain, Figure 14, and cognitive domain models [A1], Figure 15, describe this new condition:

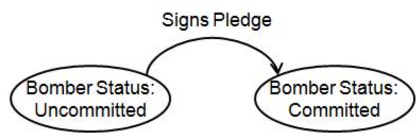


Figure 14: Information Domain Model

Figure 15 is a cognitive space model using a two-variable description of emotion. A way of describe emotion is to use a variable to describe the range from "feeling good" through "feeling bad" and use another variable to describe "how much". This particular Cognitive model shows that the suicide bomber feels better after signing the pledge and is calmer:

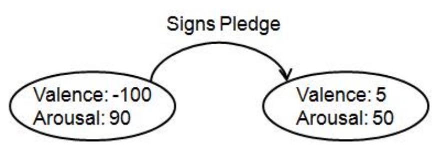


Figure 15: Cognitive Domain Model

It is a fact that societal, cognitive, and information domains depend upon the existence of physical objects, thus a complete operation in the physical domain may seem to be an emotionally, politically, and economically attractive solution. However, complete operation in the physical domain is difficult and history has shown that cognitive domain quiescence is a long-term stable state in the C2 problem space.

## Structural and Situational Complexity

At a high-abstraction level where various real world scenarios are summarized to the word "problem", it is a fact that increase in problem size causes corresponding increase (sometimes exponentially) in C2 tool processing time, increase in situation awareness uncertainty, and increase in decision time. A reason for C2 abstract problem space complexity is due to the number of variables that must be included to describe a problem space. Each additional variable added to a model potentially induces a problem space that grows disproportionally large and results in excessive resource consumption during the construction, analysis, and utilization of the model. In C2 research, it is tempting to apply problem-solving methodologies to a highly abstracted problem space. However, the folly of working on an abstracted problem space is the assumption of an all-representing problem space. It is not possible to represent an all-encompassing problem space. However, it is more engineering doable to represent a particular problem in sufficient fidelity for the purpose to solve specific problems. Characteristically, the useful realistic representation of C2 activities results in limited problem spaces and that limitation induces both *structural* and *situational* complexities, as shown in Figure 16 below.

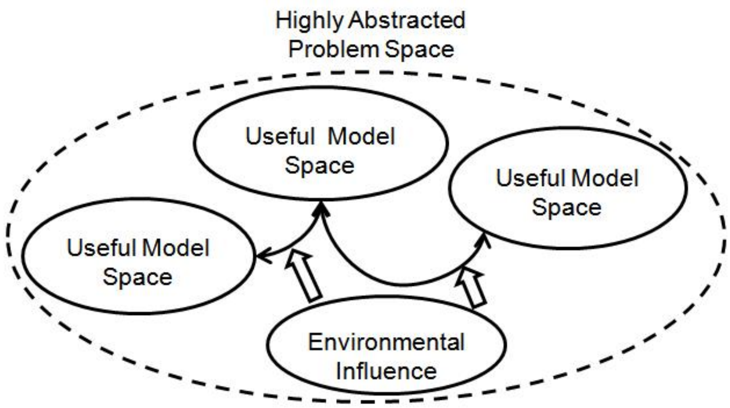


Figure 16, Structural and Situational Complexities

Structural complexity is an attribute of the solid ovals inside the all-encompassing problem space. Structural complexity is induced by the perspective variables. For example, in the recent Haiti earthquake relief effort, these variables may be transportation-related in coordinating sustenance and medical resources, with the location of victims. A modern C2 problem has situational complexity induced by the limited representative power of a problem space. The solutions are transportation-centric and space representation would be less useful if rioting occurs. This is situational complexity: necessity to re-formulate limited problem space representation due to unforeseen situational factors. The situational complexity is particularly problematic when situational forces are intelligent and adversarial. Reasonable higher situation awareness and response are indicated for C2 activities with such situations. However, the CAS representation is still applicable to these scenarios.

## Implementation of CAS

Below is an additional example of CAS across the physical, information, and cognitive domains. When an action has been carried out, the direction of the edge indicates the original and resultant states. For example, Figure 17 is a CAS model of a customer purchasing a coat for a child:

(Coat location: on counter) ---> pay cashier ---> (Coat location: on counter)

Figure 17, Physical Domain Effect

We note that the effect in the physical domain is none by the action of the customer paying the clerk for the coat.  However, the effects on the information, cognitive, and social domains are not constant, as shown in Figure 18. Each row begins with the domain, followed by the particular domain variable and its value before the action, followed by the domain variable and its changed value:

Information: (Coat ownership: store) ---> pay cashier ---> (Coat ownership: customer)
Information: (Clerk commission: $50) ---> pay cashier ---> (Clerk commission: $55)
Cognitive: (Customer Emotion: -25, 60) ---> pay cashier ---> (Customer Emotion: -20, 50)
Social: (Economic Activity: 1,000.01) ---> pay cashier ---> (Economic Activity: 1,000.02)

Figure 18, Information, Cognitive, Social Domains Effects

Each row of Figure 18 is a step from a state to another state. All three battle domains can be combined to form a single problem space or the problem space can be segmented into sub-domains, such as in this example which the space has been segmented to Physical, Information, Cognitive, and Social domains. It is likely that segmented domain in problem space analysis generate less complex problem space than combined approach. This assertion need to be further investigated by real world experimentation.

## Summary

The Common Agility Space (CAS) can function as a common information domain level data structure that provides a unified and evolvable construct for C2 processes, tools, and users. We have demonstrated an analysis using the CAS framework that showed how choosing an approach to solve a problem was significant in problem space growth. The CAS framework is likely to also be valuable in additional C2 activities and analysis. CAS's representative power spans the physical, information, cognitive, and societal domains and its representative capability will likely reduce inconsistencies between strategic, operational and tactical activities. CAS can improve consistency and C2 effectiveness. This construct is a connection method for existing command and control concepts such as:

- Situation Awareness identifies states in the CAS.
- Effects-based Operation selects states in the CAS.
- Resource spent on C2 activities can be measured as the paths lengths between CAS states.

We have explained the construct through an example. However, issues concerning the instantiation of the problem space have been left for future investigation.

C2 activities seek to transition a system to a stable state in an environment that contains multiple influential forces [L1]. These forces permeate the environment, when viewed through the battle-space perspective, generate activities in multiple battle-space domains which require response and decrease resources needed to stabilize a system. These forces can be known or unknown, natural or intentional, sporadic or pandemic. Thus an empowered C2 organization needs to assure its strategic commands are encompassing to achieve its goals, its operational controls are responsive, and the ramifications of its tactical activities move the system towards desired effects. The CAS construct is able to represent all three levels of C2 activities. The endeavor of stabilizing a system involves using command and control methodologies and tools to negotiate in the physical, informational, cognitive, and societal domains.

## References

[A1] David S. Alberts, John J. Garstka, Frederick P. Stein, "Network Centric Warface: Developing and Leveraging Information Superiority, 2nd Edition (Revised)", CCRP Publication, 2000.

[A2] Simon R. Atkinson, James Moffat, "The Agile Organization," CCRP Publication, 2005. (bookmark pg. 46)

[B1] Brian Beavis and Ian M. Dobbs, "Optimization and Stability Theory for Economic Analysis", ISBN 0521336058, 9780521336055, Cambridge University Press, 1990.

[B2] Carl H. Builder, Steven C. Bankes, and Richard Nordin, "Command Concepts: A Theory Derived from the Practice of Command and Control", ISBN 0-8330-2450-7, RAND, 1999.

[C1] James Samuel Coleman, "Foundations of Social Theory", ISBN 0674312260, Harvard University Press, 1990.

[D1] Peter Daszak, Andrew A. Cunningham, and Alex D. Hyatt, "Emerging Infectious Diseases of Wildlife - - Threats to Biodiversity and Human Health", Science, vol. 287, no. 5452, pp. 443 – 449, 21 January 2000.

[E1] Walter Enders and Todd Sandler, "Terrorism: Theory and applications", Handbook of Defense Economics, Chapter 9, vol. 1, pp. 213 – 249, Elsevier B. V., 1995.

[E2] Mica R. Endsley and Daniel J. Garland, "Situation Awareness: A?nalysis and Measurement", ISBN ISBN 0805821341, 9780805821345, Lawrence Erlbaum Associates, 2000.

[G1] J. Gong, G. Wahba, D. R. Johnson, and J. Tribbia, "Adaptive Tuning of Numerical Weather Prediction Models: Simultaneous Estimation of Weighting, Smoothing, and Physical Parameters", Monthly Weather Review, vol. 126, iss. 1, January 1998, pp. 210 – 231.

[L1] Bruno Latour, "Reassembling the Social: An Introduction to Actor-network-theory", ISBN ISBN 0199256047, 9780199256044, Oxford University Press, 2005.

[M1] James Moffat, "Complexity Theory and Network Centric Warfare", CCRP, 2003.

[N1] Nils J. Nilsson, "Artificial Intelligence: A New Synthesis", ISBN 1558604677, 9781558604674, Morgan Kaufmann, 1998.

[S1] M. Sethumadhavan, "Discrete Mathematics and Its Applications", ISBN ISBN 8173197318, 9788173197314, Alpha Science International Ltd., 2006.

[S2] Edward A. Smith, "Complexity, networking, and effects-based approaches to operations," CCRP Publications, 2006.

[S3] Edward A. Smith, “Effects Based Operations,” CCRP Publications, 2003.

[Z1] Weixiong Zhang, “State-Space Search: Algorithms, Complexity, Extensions, and Applications,” ISBN 0-387-98832-7, Springer-Verlag, 1999.