12th ICCRTS

"Adapting C2 to the 21st Century"

Title of Paper:

Mapping Network Centric Operational Architectures to C2 and Software Architectures

Topics

C2 Technologies and Systems; C2 Metrics and Assessment; C2 Concepts, Theory, and Policy

Authors: Jack Lenahan, Imagine-One Corporation
Phil Charles, Command Chief Engineer,
Space and Naval Warfare Systems Center Charleston,
South Carolina
Rebecca Reed, SRC Corporation, Don Pacetti, ManTech Corporation, Mike Nash, SPAWAR Charleston
POC: Jack Lenahan
Organization: Office of the Chief Engineer
Space and NAVAL Warfare Systems Command
Charleston, S.C.
Address: P.O. Box 190022
N. Charleston, South Carolina: 29419
Phone: 843-218-6080
Email: John.Lenahan@Navy.mil

Mapping Operational Architectures to C2 and Software Architectures

Agenda

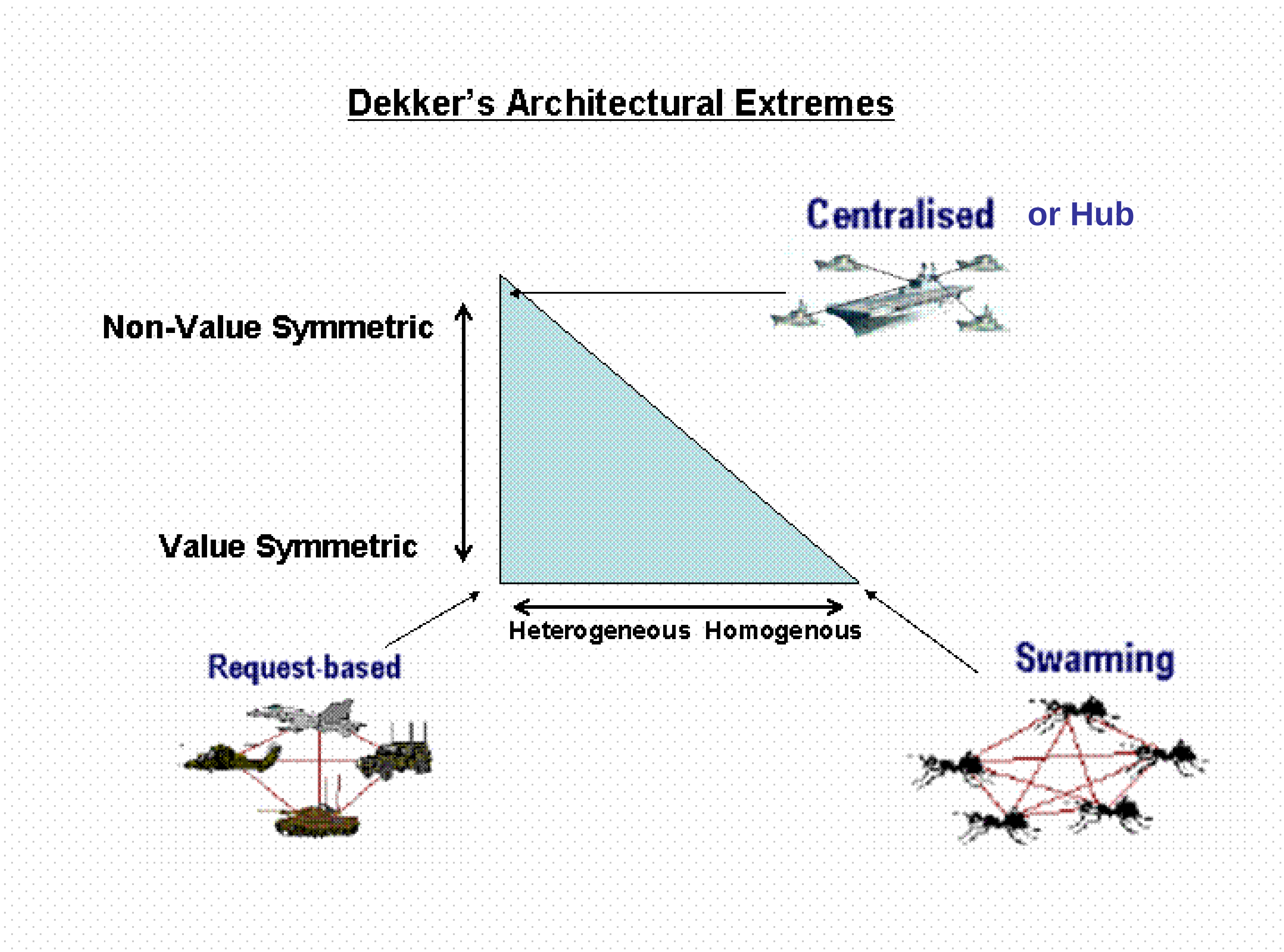
- Introduction
- Definitions
- Evaluation Process
- Context
- Hypothetical Scenario
- Conclusions

Introduction

- **This paper describes the results of a gedanken experiment. Thought experiment methodology is a priori, rather than empirical, in that it does not proceed by observation or physical experiment.**
- **Thought experiments are well-structured hypothetical questions that employ "What if?" reasoning.**
- **We are attempting to fuse, we believe, for the first time, an analysis of Network Centric Operational Architectures, Command and Control Strategies,and Software Architectures**
- **By evaluating variances in C2 stratgey and Software Architectures against a steady state operational configuration, we believe that we can shed some light on the operational consequences of fused aspects of NCW theory. We hope that this effort leads to a more thorough analysis of the actual operational impact of sudden simultaneous deviation from traditional command and control at the same time we introduce service oriented architectures.**
- **In our case, we wish to evaluate the proposed architectural taxonomies of Dekker as a possible set of operational baseline configurations with respect to their relationships to command and control models (C2) and software architectures. The Dekker architecture types which will be evaluated are hub request, hub swarming, request based (without a hub), emergent swarming (leaderless), hierarchical swarming, orchestrated swarming, and distributed swarming (leaderless). For this gedanken experiment, a swarm is assumed to have the properties of swarm intelligence normally associated with Particle Swarm Optimization (PSO).**

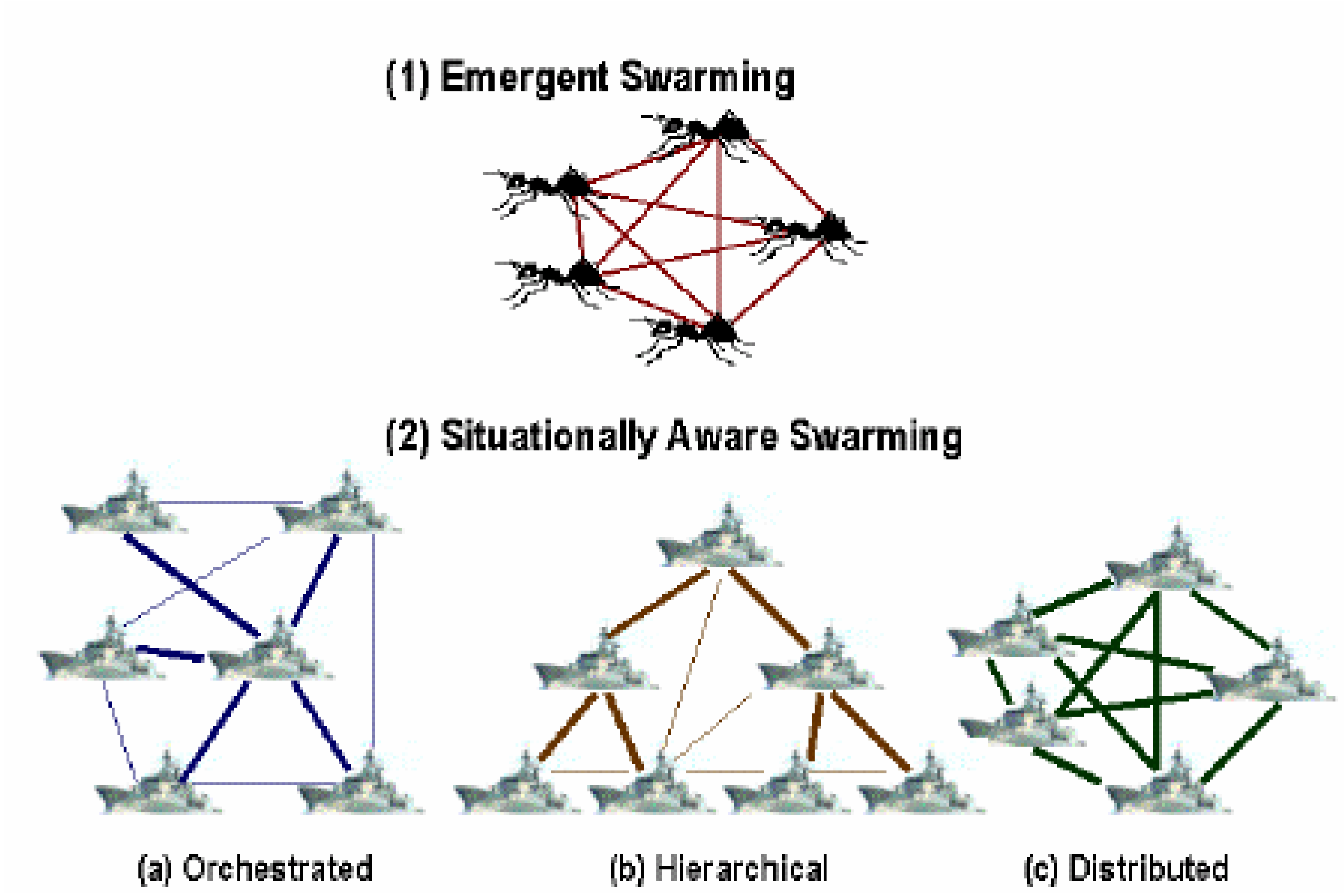
Introduction Continued

- If one were to view our model in a traditional IDEF process model context, we are simply expanding the modeling of the controls (C2 Strategy) and the mechanisms (assets and Software Infrastructure) to enrich the value of future simulations. Thus, in effect, we are proposing a 3 dimensional framework for assessing multiple aspects of NCW.
- Our gedanken experiment results show that the configuration of assets and how they were organized (commanded and controlled) actually increased their collective capabilities given an optimized hybrid SOA, MOMS (Message Oriented Middleware), and Agent Based software infrastructure.
- This means that any capability portfolio analysis or competency assessments which only focuses upon individual asset contributions, fails to account for the behavior of a team or the possibility of “collective swarm intelligence”.
- This almost by definition will lead to procurement decisions detrimental to the basic capability of the DoD.



Modified by this author from Dekker's original

Swarming Architectures



Dekker Architectures Evaluated

- **The Dekker architectures evaluated were**
  - **Hub request - A Hub Request Architecture is a configuration characterized by “a single high-value central “hub” node, surrounded by a cluster of nodes of lower value. The central “hub” provides services of such high value that the force cannot operate effectively without it. The “hub” is therefore what Clausewitz called the “*center of gravity... on which everything depends*”.**
  - **Hub swarming - a hub swarm model uses one of the nodes as a centralized command and control “leader”. Hub Swarm architectures involve a mix of nodes of different kinds and values. Such a mix arises particularly in a Joint force, and involves mixing elements of all the other types of NCW.**
  - **Request based (without a hub) - A request based architecture, defined as the combination of fully value-symmetric and heterogeneous forces, is a collection of pure specialists, all different, but all of equal value. Each node does only a few things, and does them extremely well. Since military operations require multiple coordinated tasks, each node must call on many others to perform tasks that it cannot do**
  - **Emergent swarming (leaderless) - Emergent Swarming occurs in nature among insects such as ants (Gordon 1999): “The basic mystery about ant colonies is that there is no management. A functioning organization with no one in charge is so unlike the way humans operate as to be virtually inconceivable. There is no central control. ... No ant is able to assess the global needs of the colony, or to count how many workers are engaged in each task and decide how many should be allocated differently. The capacities of individuals are limited.**

Dekker Architectures Evaluated Continued

- **Hierarchical swarming - Hierarchical Swarming is closest to the traditional military C2 architectures, and this is because it represents an extremely good solution for dealing with complex problems. In Hierarchical Swarming, the nodes are organized into a hierarchy. In the event of nodes being lost, the hierarchy is maintained by promoting other nodes. Situational awareness information is fused going up the hierarchy, and at the same time, low-level tactical detail is dropped out. This means that the commanding node gets the “big picture” situation awareness that it needs. This simplifies the situational awareness fusion problem and avoids over-straining the information fusion capability of the nodes. The commanding node then produces a “big picture” plan (often called “intent”). This is passed down the hierarchy, and tactical detail is added by subordinate nodes. This avoids over-straining the planning capability of nodes.**
- **Orchestrated swarming - In Orchestrated Swarming, one of the nodes is chosen as a temporary “leader.” In the Centralized Architecture, the C2 node was the node best equipped for command and control activities, but in swarming architectures, all the nodes are identical. The choice of “leader” is therefore made on the basis of suitable position, current combat situation, or other transient factors. This approach is sometimes used in Special Forces teams, where members can, if necessary, take over command from the nominal commander. Sensor data is sent to the “leader” node, where it is fused to produce an integrated situational awareness picture and an integrated plan of action. These are then broadcast to the other nodes. If the leader is unable to continue for any reason, the nodes agree on a replacement, which takes up where the previous leader left off. This approach limits network traffic, but it puts great stress on the C2 capability of the leader**
- **Distributed swarming (leaderless) - Distributed Swarming16 has no “leader” role, and all decisions are made through consensus. Situational awareness is handled by all nodes broadcasting their sensor information, so that every node builds up an individual situational awareness picture.**

C2 Approaches Evaluated

- The Command and Control Approaches Evaluated Were:
  - Cyclic – Chinese Army
  - Selective Control – Israeli Army
  - Interventionist – Soviet Army WWII
  - Problem Solving – American Army
  - Problem Bounding – British Army
  - Control Free – German WWII

Introduction Continued

- The software architectures evaluated were:
  - Service Oriented Architecture – SOA
  - Event Driven Architecture - EDA
  - Message Oriented Middleware - MOMS
  - Legacy Software Architectures
  - Agent Based Architectures – ABA
    - Note that the agents had learning and communication capabilities

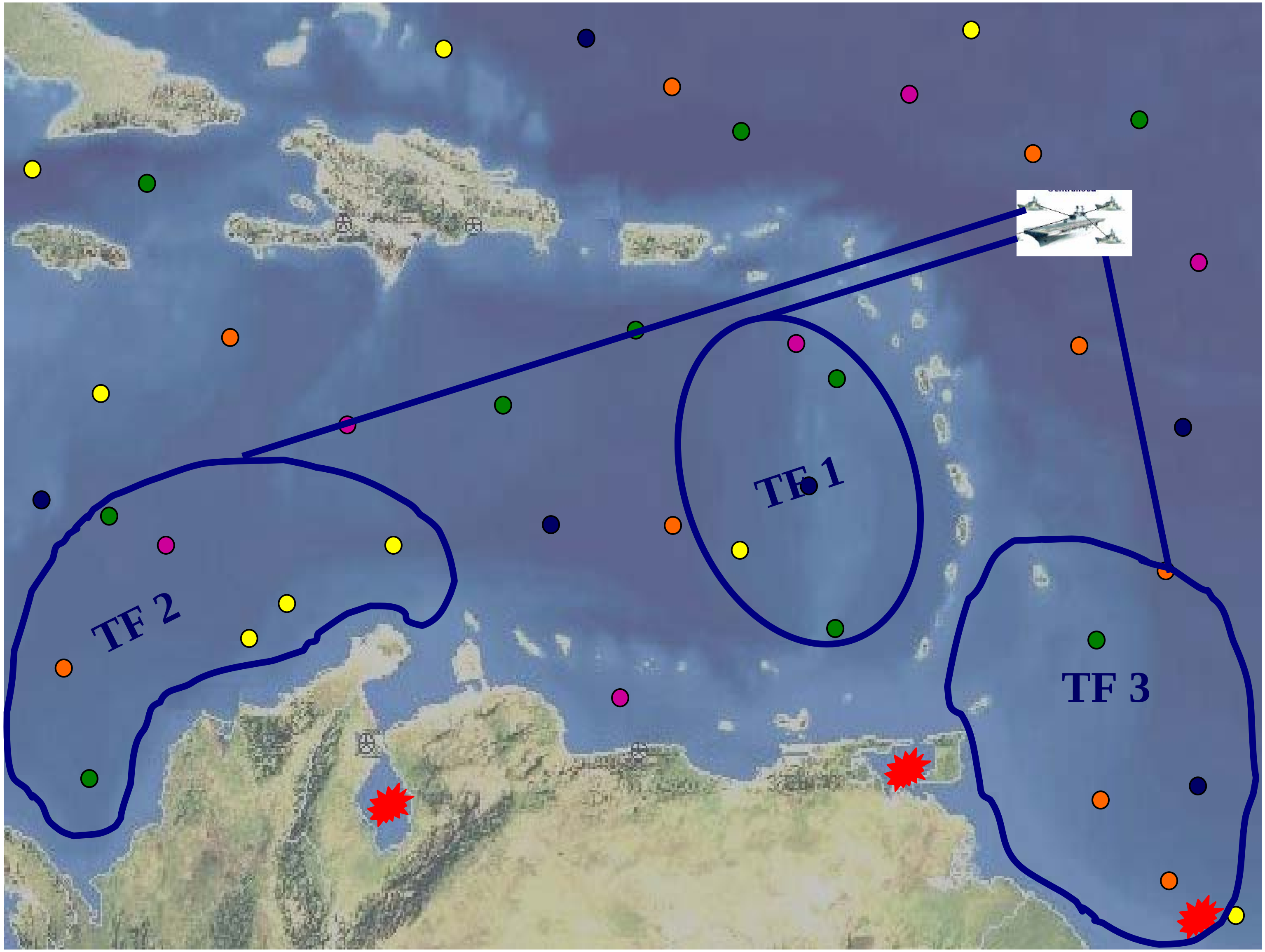
Process Followed

- Define a simple model with few variables
- Define a simple mission with clear and easy to measure metrics
- For each of the Dekker models, vary the command and control and software architecture models and measure

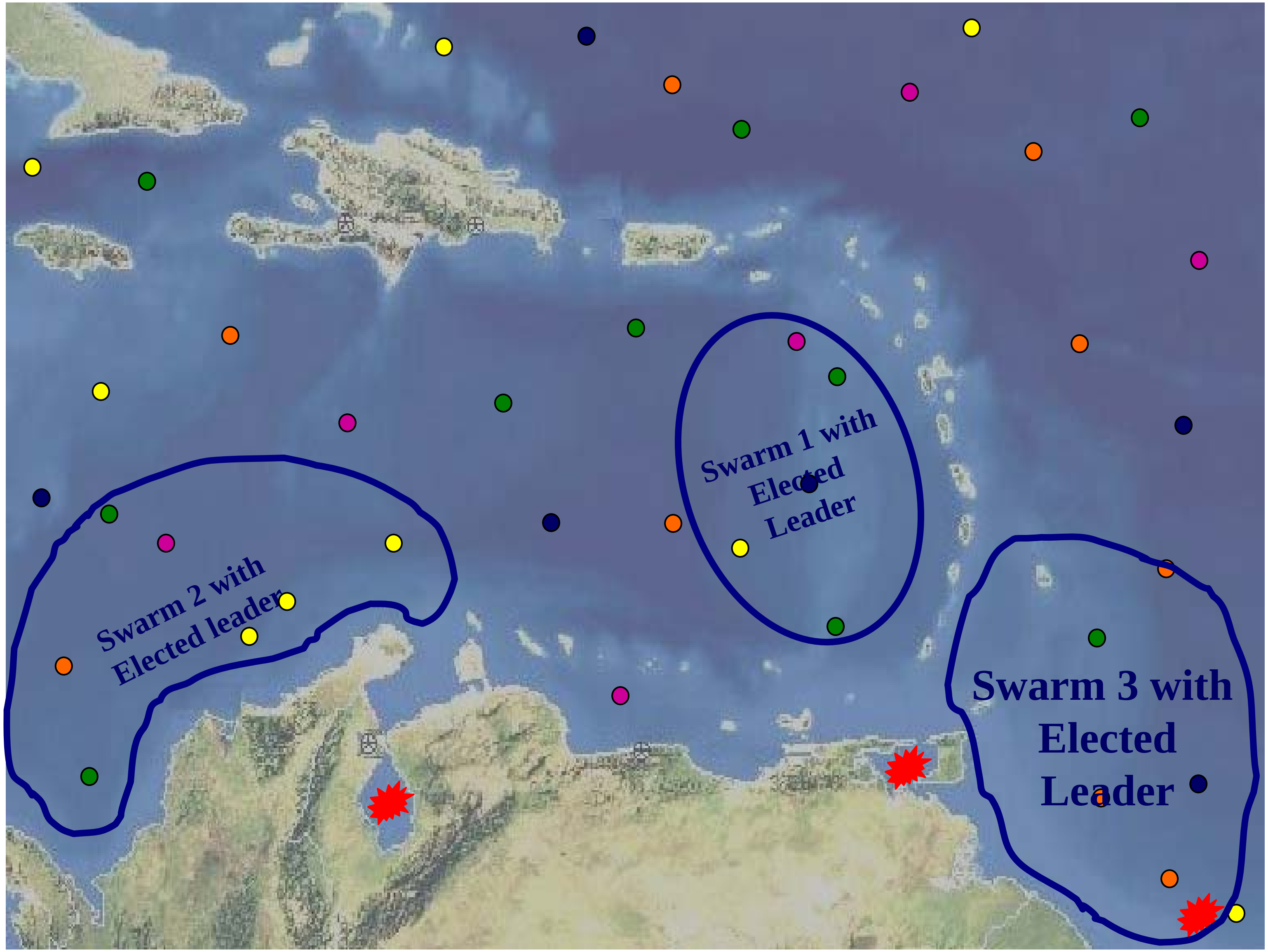
Context – Hypothetical Mission to find a Missing Plane

- Assume that we have a set of drones available on a sensor grid
- Each of the drones are fueled and available for tasking
- The drones receive their tasking via GIG Sensor Grid Communications or directly from a “leader drone” in Hub models
- Assume that there are only 3 types of drones available:
  - For this simple example, all drone types have equivalent sensors & range
  - Each drone of the first type contains onboard artificially intelligent software agents capable of planning a search and rescue mission
  - Each drone of the first type can be appointed as a command node and issues search pattern commands to the non command nodes
  - Each drone of the second type cannot plan a mission and can only follow orders
  - Each drone of the third type is used for protection only. Thus it cannot be used in searches in hub type architectures
  - Depending upon the architectural configuration, the onboard agents will be able to communicate with each other or only to a leader.
  - Drones of all types can communicate with the sensor Grid or each other
- The sensor grid contains an adjudication agent which will deconflict concurrent or competing asset requests. This agent was not implemented or its impact on finding the missing plane analyzed for this effort due to staff resource constraints. This is mentioned only to complete the sensor grid description since a given a real set of sensors, some task and sensor request adjudicator will be necessary.

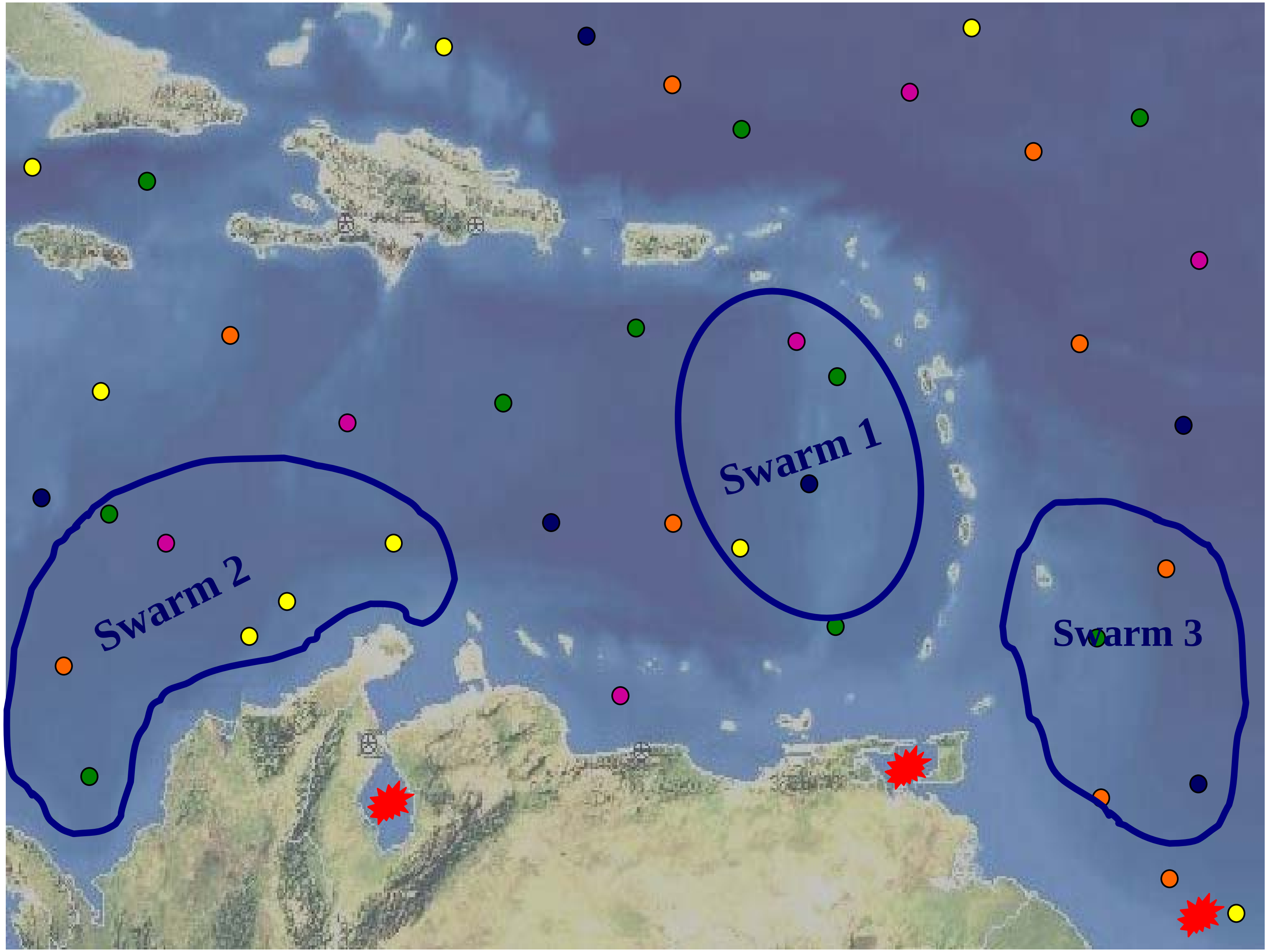
Hub or Centralized Architecture Behavior
Problem Solving (US) C2 & SOA, MOMS Legacy and Global Situational Awareness Maintained by Hub



Orchestrated Swarming Behavior – German Control Free, SOA,MOMS, Agents –Collective Global Situational Awareness Available Through the SOA via Each Swarm's Publishing up the Chain of Command



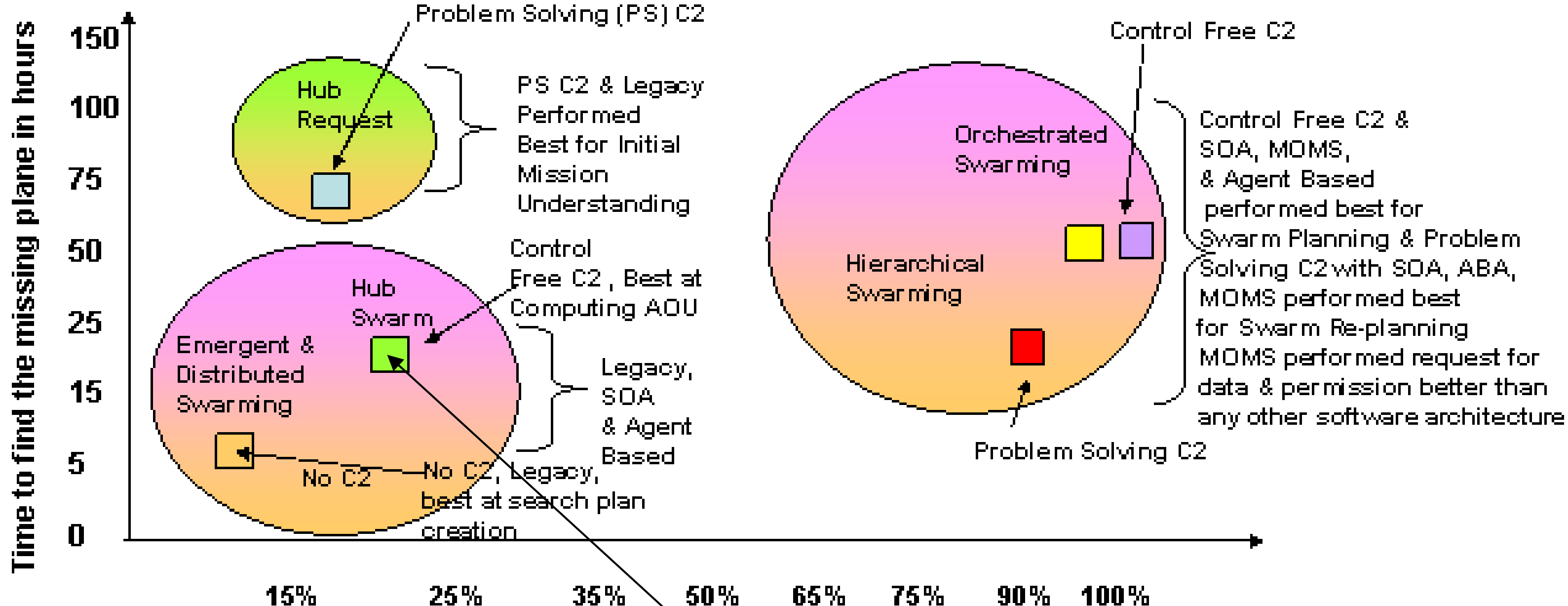
Distributed Swarming Behavior – Leaderless with Some Assets Choosing Not Play and Others Randomly Joining Different Swarm Groups – Swarm 3 Misses Target



Metrics

- The metrics which will be used to judge each configuration are:
  - Time for the leader drone to process the mission request and “understand it”, for hub or leader based architectures.
  - Time for the “swarm” to process a mission request and “understand it” in non-hub models
  - Time for a leader node to create a search plan.
  - Time for a swarm to create a search plan.
  - Time to compute the area of uncertainty by a leader
  - Time to compute the area of uncertainty by a swarm.
  - Time to determine the search plan for each individual drone by a leader
  - Tine to determine the search plan for individual drones if calculated by the swarm
  - Time for requests to be processed from each drone to the leader
  - Time to re-plan by a leader model if first searches are unsuccessful
  - Time to re-plan by a swarm if first searches are unsuccessful
  - Time from mission start until mission completion (missing plane found)

Solution Space of Most Reliable Configurations of Operational Architectures, Command and Control Models, and Software Architectures. The color coded tasks in each indicate the solution space region where optimal performance for that task was achieved given a particular C2, Dekker, and Software configuration

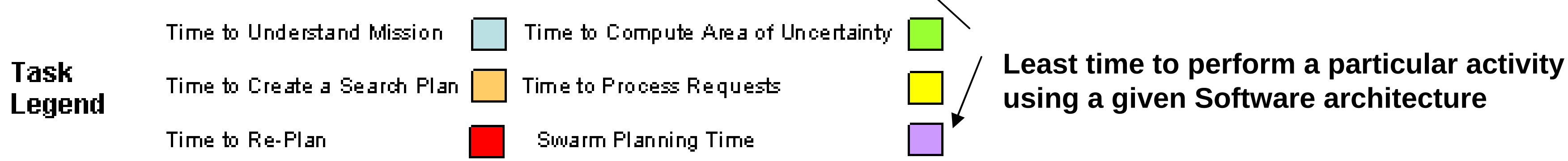


The x axis represents the probability that the missing plane will be found at the same location repeatedly.

Results:

Emergent & Distributed Swarming found the plane the fastest but least consistently. They also had the fastest time to process requests because there was no C2 in place eliminating approvals from a chain of command.

The hub swarm with control free C2 and Legacy, SOA, ABA, excelled at AOU computation. Hub request excelled at initial mission understanding with PS C2 and a Legacy Software Architecture - Hierarchical and Orchestrated Swarming found the plane most consistently using SOA, MOMS, & ABA software architectures. Orchestrated Swarming excelled at swarm planning and request processing while hierarchical swarming excelled at swarm re-planning using a PS C2 model. The hubs were also inconsistent because they had to assign a protector drone and thus had fewer search assets. Thus, in the end planning & C2 mattered in arriving at consistent repeatable results



Observations

- No single command and control model worked (optimized performance) for the mission as a whole
- This means that the individual tasks responded better under different command structures.
- No single software architecture achieved superior results for the mission as a whole

Results

- **By definition, operational Hub architectures which required a protector drone had at least one less search asset. Thus, these Hub models were less successful in terms of time to find the missing plane than leaderless models requiring no protector drone.**
- **Orchestrated swarming consistently performed better than any other operational architecture configuration given the simple scenario of finding the missing plane in a fixed time period.**
- **The primary characteristics that we were looking for was consistency of the discovery of the missing plane without a re-planning cycle and the elapsed mission time. In some placements of the missing plane, distributed and emergent swarming (both leaderless) did actually find the missing plane quicker and without re-planning, but not consistently.**
- **In the orchestrated model, the election of the leader did not preclude individual drone initiative and communications between all the other nodes was also enabled. Disabling inter-nodal communications had an adverse impact on all of Dekker's configurations.**

Results Continued

- Dekker's Orchestrated Swarm Architecture using a hybrid software architecture of SOA, ABA, and MOMS, configurations performed best at the mission level
- At the task level
  - Computation intensive tasks (planning and AOU computation) performed best on legacy systems for hub architectures
  - Computation intensive tasks on swarm architectures outperformed legacy through the use of intelligent Agent Based Architectures (GA & ANN based) & Particle Swarm Optimization
  - Message intensive tasks and configurations performed best under the MOMS architecture for both hubs and swarms

Results Continued

- The leaderless C2 models also resulted in less than optimal resource utilization resulting in more frequent re-planning and longer times to successfully complete a search.
- The leaderless swarm models repeatedly duplicated failed search patterns causing excessive amounts of re-planning
- More likely to fall for a ruse

Results Continued - A few comments concerning capability portfolio management of assets and organizational competency

- **It is worth noting that the individual drone assets did not change in capability. This is an obvious but often overlooked aspect of NCW research.**
- **The configuration of the assets and how they were organized actually increased their collective capabilities.**
  - Orchestrated swarming can therefore be said to have exhibited an emergent capability of consistently finding the missing plane in time, this capability was not exhibited by the other configurations to the same degree. Yet all that changed was the organization and how they communicated, not the original capabilities of any single asset. It may be fair to state that indeed individual competency and capability increased by the re-organization of the assets and the methodology of permitting either more or less practical levels of individual freedom of action.
- This means that any capability portfolio analysis or competency assessments which do not take collective emergent behavior into account are at best going to cause budgetary overruns and at worst make procurement decisions to the detriment of the basic capability of the United States Military.