Resilient C2 in the A2/AD Environment

Martin O. Hofmann, Katherine Guo, Kevin Spivey, Rachel Hingst
Lockheed Martin Advanced Technology Laboratories
3 Executive Campus, Cherry Hill, NJ 08002, USA
Martin.Hofmann@lmco.com, +1 856 792 9711

Abstract

In contested environments, where communications with the centralized Air Operation Center (AOC) are denied or degraded, the forward-located Distributed Control Nodes (DCNs) will need to assume the AOC's planning and control functions. USAF is investigating the DCN construct, since forward deployed DCNs are much less vulnerable to communications denial as they can take advantage of shorter-range, directional communications means to coordinate with each other, while the long-haul communications links to the AOC present a large, soft, high-value target to a capable adversary. The DCN construct naturally extends to coalition operations where coalition partners provide or staff forward control nodes, and where coalition forces would ideally be controlled in an integrated manner by DCNs. CONOPS featuring unified coalition operational control would enable more effective and agile employment of forces and more optimal sharing of resources. However, to achieve these benefits with only a small fraction of the AOC's manpower, expertise, and situational awareness, each DCN needs to contribute to the planning and replanning, controlling, and assessing of a comprehensive set of missions. We present an initial experiment with a decision support capability that assists DCNs' staff in allocating and sharing responsibilities for the C2 tasks (mission planning, controlling, and assessing) across a set of parallel missions. Our experiment demonstrates that an auction-based, many-to-many resource allocation technique effectively allocates and schedules C2 tasks to DCNs. Under reasonable assumptions about DCN staffing, our experiment supports the viability of the DCN construct to sustain the tempo of air operations in accordance with Commander's Intent.

1 Introduction:

Today, the Coalition Air Operations Center (AOC) provides centralized command and control of all coalition air missions. This approach has proven highly successful, but as the focus of air operations shifts from conflicts where coalition capabilities are unchallenged towards anti-access/area denial (A2/AD) scenarios, reliance on centralized control is viewed as a vulnerability. With the rise of communications jamming and anti-satellite capabilities of competitors, the *centralized control and decentralized execution* construct, which depends on rapid flow of information from the battlespace to the AOC and of control from the AOC to the battlespace, is being questioned. Therefore, mitigations and alternatives are being sought, among them the distributed control node (DCN) construct, as one realization of the new vision of *centralized command, distributed control, and decentralized execution*[1].

Changing from a centralized to a distributed C2 organization raises issues of delegating control authority, maintaining unity of effort, and employing the specialized skill mix of staff at small DCNs in the most effective manner towards the common Commander's Intent. A key issue will be to ensure that the DCNs take on the appropriate portion of the distributed C2 process for a

[1]Hostage III, Gilmary Michael; Broadwell Jr., Larry R., "Resilient Command and Control", JFQ: Joint Force Quarterly; 2014 3rd Quarter, Issue 74, p38.

subset of missions, which includes mission planning, control of mission execution, and assessment of mission effectiveness.

Individual DCNs will have fewer C2 resources, including fewer qualified people, lesser scope C2 systems and data, and shorter range communications. Some of the DCNs may not be able to operate continuously. A Control and Reporting Center (CRC), for example, will occasionally be offline while the CRC moves. Thus, allocation of C2 tasks to DCNs requires an optimal many-to-many allocation solution that rapidly adapts allocations to changing mission tasks, e.g., planning and controlling pop-up target missions or re-strike missions and reallocating C2 tasks due to DCN overload or unavailability. Counteracting this desire for operational agility are the need to prioritize C2 tasks associated with high-priority missions and to minimize allocation changes, which may be disruptive to staffs, as much as possible.

Our project, in contrast to earlier ones, does not develop an automated, distributed *mission* planning technique, but a capability to recommend optimal *allocation of C2 tasks* to human planners supported by automation at a number of DCNs.

In this paper, we present an initial experiment with a decision support capability that assists staff at DCNs in allocating and sharing responsibilities for the C2 tasks (mission planning, controlling, and assessing) across a set of concurrent, evolving missions. Our experiment demonstrates that an auction-based, many-to-many resource allocation technique effectively allocates and schedules C2 tasks to DCNs. Under reasonable assumptions about DCN staffing, our experiment supports the viability of the DCN construct to sustain the tempo of air operations in accordance with Commander's Intent.

2 Operational Vision and Requirements

Distributing the C2 functions normally handled by the centralized Coalition AOC to a set of forward deployed Distributed Control Nodes (DCNs) promises to increase resilience of C2 processes in the face of Anti-Access/Area Denial (A2/AD). USAF is investigating the DCN construct, since forward deployed DCNs are much less vulnerable to communications denial, because they can take advantage of shorter-range, directional communications means to coordinate with each other, while the long-haul communications links to the AOC present a large, soft, high-value target to a capable adversary. Wing Operations Centers (WOC), a Carrier Strike Group (CSG), a Control and Reporting Center (CRC), or an Airborne Warning And Control System (AWACS E-3A aircraft) are envisioned to serve as DCNs. The construct naturally extends to coalition operations where coalition partners provide or staff forward control nodes, and where DCNs provide integrated control of coalition forces.

Figure 1 shows a notional arrangement of DCNs in the pacific theater. Communications between DCNs is relatively short-range, but will still require the use of relays, and is, thus, not guaranteed to be continuously available. We derive the following requirements for the technical approach of a C2 task allocation scheduler from this operational vision:

- Fine-grained representation of capabilities required to perform a C2 task. For example, the task of planning a Time-Sensitive Target (TST) strike mission, requires expertise in operational planning, weapon selection, weather analysis, and airspace control among others; systems for air surveillance and air tracking; and corresponding communications

links. These capabilities need not reside in a single DCN but can be shared across DCNs, as long as communications between DCNs are available.
- Fine-grained representation of DCN proficiencies: not all DCNs are equally well qualified to perform a C2 task. Thus, an explicit model of DCN proficiency in the capabilities required by a task need to be modeled explicitly.
- A representation for constraints on task and capability allocations to DCNs, including
    - Temporal and spatial constraints, e.g., a DCN can only control missions if it has radar coverage and communications with the executing aircraft.
    - Co-allocation constraints. Some capabilities have to reside in the same DCNs, i.e., cannot be distributed, including any capability that collect or generate information and the capability to share this information.
- Graceful handling of allocation changes: changes to C2 task allocations are sometimes necessary, e.g., when a CRC moves, and sometimes desirable, when a high-priority task needs to be allocated to DCNs operating near capacity. The approach needs to maintain a near-optimal allocation that matches proficiencies to task needs and honors C2 task priorities, but without causing excessive ripple effects of re-allocations.
- Resilience to message loss between DCNs and temporary DCN unavailability.

3 Computational Model

We based the implementation of our C2 task allocation scheduler on our CLUS-STAR resource allocation algorithm (Greene & Hofmann, 2006) (Guo et al., 2016).

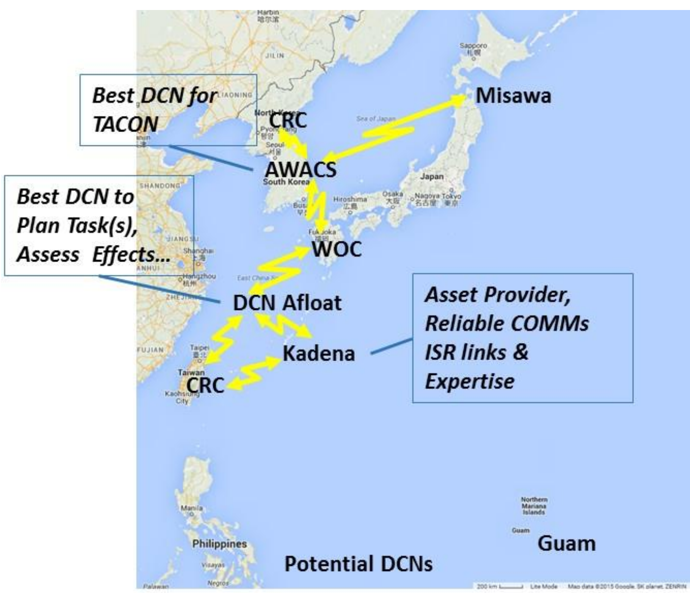

Figure 1. A notional set of DCNs and their communications links. Reachback links are considered at risk and are not shown.

In CLUS-STAR (Figure 2), we formulate the problem as dynamic, optimal resource allocation of capabilities required by tasks to a set of agents with varying proficiencies for one or more of the required capabilities. Each agent represents a resource that possesses multiple capabilities. N tasks arrive dynamically over time and are located in a given geographical region. Upon arrival of a task, agents bid to perform the task. An auction algorithm assigns the task to one or several bidding agents, specifically, selects capabilities required for the task from among the capabilities bid by the bidding agents. A penalty is incurred if a task remains unassigned or is later dropped before it is executed. The completion of a task incurs a cost and also earns a reward for the agent that is specific to the agent. The goal of the optimization problem is to maximize net revenue (reward minus cost) over time for all tasks, regardless of which agents executes the tasks. In contrast to a competitive auction, agents do not maximize personal reward, but contribute to the team award. Since agents do not compete with each other, a simple bidding technique is appropriate, where agents simply bid actual cost/reward, without regard to competing bids.

The single auctioneer agent may appear to present a weakness, but we typically host the CLUS-STAR algorithm on a multi-agent platform, such as our Extensible Mobile Agent Architecture (EMAA)[2] agent framework, the Java Agent Development Framework (JADE), or an equivalent framework. These platforms provide mechanisms for hot spares and fail-over. They support formation of sub-groups of agents that perform localized auctions when the communications network splits into multiple sub-nets (Sheu et al., 2010).

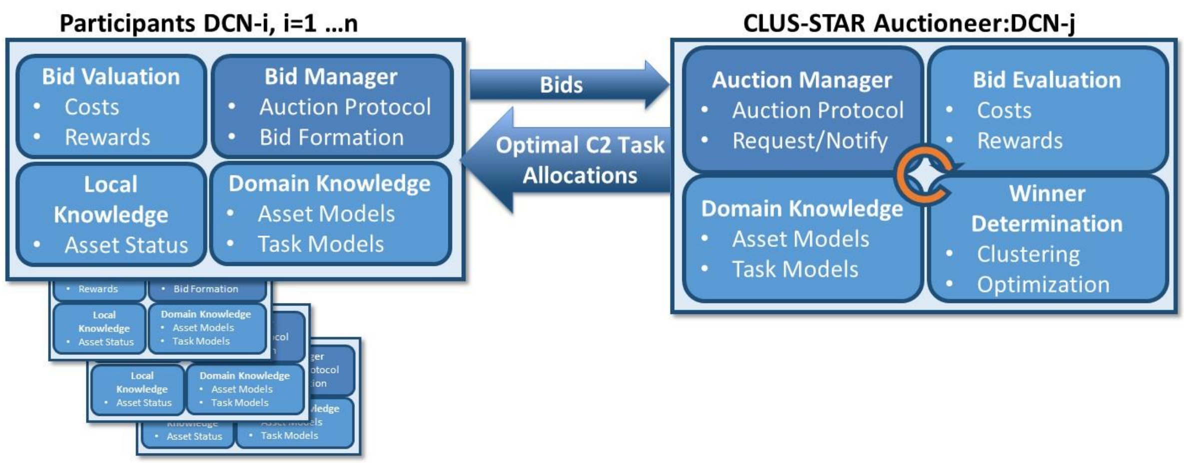

Figure 2: One or more DCNs serve as auctioneers, evaluate bids from peer DCNs, and optimize C2 task allocations across DCNs. Bidding DCNs bid for all the tasks they are able to perform given their local resources.

CLUS-STAR has a unique combination of characteristics that make it specifically relevant to the C2 task allocation scheduling problem:

(1) It handles tasks that require cooperation by multiple capabilities hosted at multiple DCNs, supporting many-to-many assignments when spatio-temporal constraints and communications availability allow.
(2) CLUS-STAR will break commitments in favor of higher-priority pop-up tasks, but only when the overall benefit outweighs the cost of the change and the loss of reward from the earlier assignment. This limits changes to a few worthwhile cases and increases scalability due to reduced churn.
(3) CLUS-STAR has low and very localized communications requirements and relatively low computational complexity due to its single-stage negotiation. It implements winner determination with a polynomial time, constrained-clustering algorithm.
(4) Supported by bidding proxies, our CLUS-STAR peer-to-peer solution is highly resilient to communications network degradation. A proxy may bid on behalf of the agent it represents when that agent is unreachable but it is expected to regain communications before the task is scheduled to be executed. If communications remain lost, the auction for the task is reopened.

[2] http://www.atl.external.lmco.com/programs/EMAA/

For the application described in this paper, we extended our CLUS-STAR algorithm to allocate portions of resources to a task. Apportioned resources model the ability of people to self-manage their attention among multiple tasks, the ability of systems to execute multiple functions, and the ability of communications links to transmit on multiple channels. Another extension for this application implemented on the current CLUS-STAR version supports reassignment of partially completed tasks, as may be required when a CRC moves.

An attractive feature for coalition operations is that, by virtue of our auction technique, DCNs do not need to share their full state information with a centralized optimization algorithm hosted by a coalition partner. DCNs need only respond to a new request for bids by offering selected capabilities they want to make available.

3.1 Comparison to other Work

Karlsson et al. (2007) summarize the arguments for applying a market-based approach to resource allocation. Here, we paraphrase three we found important:

- Market-based approaches represent a convenient abstraction that cover a wide variety of resource optimization problems
- Market-based approaches provide a natural decomposition of the computational model
- Market-based approaches are allow for dynamic addition and deletion of agents

In the class of market-based approaches, CLUS-STAR's distributed auction approach for resource assignment is related to the stochastic clustering approach work by Zhang et al. (2012) but uses greedy moves for faster convergence. Thus, CLUS-STAR convergence can be expected to be faster, as demonstrated by solving 200-agent problems in seconds on a standard PC, but SCA explores a larger solution space. Also, they do not describe how the technique extends to heterogeneous tasks and agents.

Like work by Lumezanu et al. (2008), CLUS-STAR can anticipate resource congestion and adapt allocations accordingly with the added ability to deal with probabilistic and worst case future demand and asset failure without prior knowledge of future tasks. Our predictive approach deals with probabilistic and worst case future demand and asset failure without prior knowledge of future tasks.

Moghaddam et al. (2013) present cohesion of capabilities/services as an additional concern to the combinatorial auction model to reflect higher reliability of multiple services provided by a single agent versus a set of individual services provided by multiple agents. Our recent enhancements model this situation with soft constraints.

Work by Mauadi et al. (2011) on multi-agent resource allocation using combinatorial auctions, is similar to CLUS-STAR, in that it formulates a multiple-unit knapsack problem and uses a similar task and resource model – time constraints, cost (including distance, time required) etc., but considers neither dynamic re-allocation nor demand anticipation.

Amador, et al., (2014) present a multi-agent task allocation algorithm with spatial and temporal constraints similar to CLUS-STAR. Constraints are used to ensure that all of the agents that cooperate on a task arrive at the task location at the same time, however, not during the auction process but afterwards by each agent to reschedule their assigned tasks to honor the temporal constraints. The initial allocation ignores spatio-temporal constraints and schedules each agent's tasks by re-ordering tasks, e.g., moving its next scheduled task before the shared task if the agent were to arrive too early to a shared task. The algorithm allows interrupting execution of low-priority tasks. It applies only to homogenous agents.

Compared to consensus protocols, the primary concern of the CLUS-STAR auction is optimization. Tolerance of communications faults (dropped messages) is handled via proxies. It does not require iterations because it uses a single announce-bid-award auction round, and is, thus, fast and efficient. The primary concern of consensus algorithms is agreement among a number of agents (reaching the same conclusion) even in the face of faults of a minority of agents. The consensus approach is useful in fault-tolerant distributed computing applications. Consensus works correctly despite a number of failed nodes, even malicious nodes. On the other hand, optimality of the solution has to be ensured by additional means external to the basic consensus protocol. Since it does not use an auctioneer to run the optimization step, optimization requires multiple iterations over proposed allocation solutions, typically using a gradient function.

Fault tolerance, a strength of consensus techniques, is assumed to be provided by the underlying agent framework. CLUS-STAR does not question the bid from any individual agent. Also, all agents trust that the auctioneer is honest. Our choice of a single-round auction technique as realized in CLUS-STAR is motivated by the expected application, where communications is expensive and uncertain, and where cyber security is handled elsewhere (e.g., a fake asset claiming it will perform a task). We trade off some fault tolerance against malicious agents for the benefit of rapid optimization with minimal communications needs. This is also useful for rapid re-allocation in the face of changes (pop-up tasks and agent failures). Regarding the quality of the assignments, our analysis has shown that CLUS-STAR tracks the theoretical performance bounds very well.

4 Experiment

We performed an initial experiment with our C2 task allocation scheduler to validate that it will produce effective suggestions to assist DCNs' staff in allocating and sharing responsibilities for the mission planning, controlling, and assessing tasks across a set of parallel missions.

Our CLUS-STAR auction algorithm operates on an abstraction of tasks and resources that we configure for any specific application. We developed a domain model for a reasonably complete set of C2 tasks and a number of notional DCNs. The model specifies the tasks, the capabilities that a task requires, the assets (DCNs in this case) and the capabilities that they provide. CLUS-STAR then negotiates and updates allocations that make optimal use of all available capabilities across the DCNs. Allocating the capabilities required by a tasks across multiple DCNs will, in general, allow more tasks to be fully allocated than if tasks were constrained to be completely handled by a single DCN. CLUS-STAR provides hard and soft constraints to ensure that interdependent capabilities, such as creating and disseminating a mission plan are allocated to the same DCN.

4.1 Domain Formulation

We contracted PatchPlus to support development of a notional set of DCN capabilities and air missions, which the team turned into a configuration file for the CLUS-STAR algorithm.

Table 1 lists the subset of C2 tasks modeled. These tasks are representative of typical C2 activities required in air operations planning. We modeled two types of C2 tasks that impose different requirements on the C2 task allocation solution. Planning, controlling, and assessing Strike, Close Air Support (CAS), and Defensive Counter Air (DCA) missions can typically be allocated and scheduled in advance, while planning, controlling, and assessing Time-Sensitive Target (TST) prosecution often takes precedence over routine C2 tasks and has to be dynamically inserted into the workflow of the DCN staff.

Table 1: C2 Tasks included in the model

Plan TST	Plan Strike	Plan CAS	Plan DCA
Control TST	Control Strike	Control CAS	Control DCA
Asses TST	Assess Strike	Assess CAS	Assess DCA

Table 2 lists the three types of capabilities included in our model, the people (skills and authority), communications links, and tools or documents required in the performance of the C2 task.

Table 2: Notional but representative capabilities included in the model

People	Communications	Tools/Docs
Command Element	VOIP	VOIP
Ops Planner	JWICS	Chat
ISR Planner	SIPR	TBMCS
Spectrum Planner	Link-16	AOI Coverage (Radar/COP/etc.)
Airspace Control	UpChannel – AOC	JTIDS
PED GEOINT	Lateral- other C2 Nodes	ROE
PED SIGINT	DownChannel – Unit	TST Matrix
Intel	AC-1	ATO
CSAR	Tactical Data Links	DCIDE
Air Surveillance/ Tracking	SATCOM	TAW-like capability
Weapons Officer	Radios	Joint Fires capability (JADOCS)
Targeteer		Unit level Intel
Weather		No Strike List

We configured each of the DCNs in the scenario with a subset of capabilities, which would force the SPICE allocation solution to creatively combine capabilities from multiple DCNs. Table 3 shows the People capabilities of a notional Wing Operations Center (WOC). Our simplified model only specifies a nominal proficiency or “1.0”. Communications and Tools/Docs capabilities are specified by a binary code: available or unavailable. Other DCN types (CRC, AWACS, etc.) are staffed and equipped differently and, thus, have different capabilities (not shown).

Table 3: People capabilities available at a notional WOC

Capability-People	#	Proficiency
Approving authority	2	1.0
Ops Planner	2	1.0
ISR Planner	0	1.0
Spectrum Planner	0	1.0
Airspace Control	0	1.0
PED GEOINT	0	1.0
PED SIGINT	0	1.0
Intel (Ground and air threats)	2	1.0
CSAR	0	1.0
Air Surveillance/Tracking	0	1.0
Weapons Officer	0	1.0
Targeteer	1	1.0
Weather	1	1.0
…		

For each C2 task in the scenario, we specify the capabilities it requires. The configuration in Table 4 illustrates the specification of apportioned resources ("% of Person").

Table 4: Partial list of capabilities required by C2 Task "Plan TST"

Capability	Required? (Y/N)	% of Person
Approving authority	Yes	5
Ops Planner	Yes	25
ISR Planner	Yes	30
Spectrum Planner	No	
Airspace Control	Yes	10
PED GEOINT	Yes	15
PED SIGINT	Yes	15
Intel (Ground and air threats)	Yes	20
CSAR	Yes	5
Air Surveillance (feeds from radars)	Yes	20
Air Tracking (overall air picture/COP)	Yes	5
Weapons Officer	No	
Targeteer	Yes	30
Weather	Yes	5
…		

4.2 Experimental Scenario

For our initial experiment and demonstration we configured a set of notional C2 tasks. The list below shows the type, timing, duration, and priority of each of the 42 C2 tasks. We configured five DCNs with varying capabilities, a WOC, a CRC, a generic DCN, a DCGS Reachback node, and a Target Reachback node.

This initial configuration does not yet demonstrate all the features of the CLUS-STAR allocation technique. For example, the C2 nodes have enough capacity to perform all the tasks. With a higher task load, CLUS-STAR would favor high priority tasks over low priority ones.

1. Plan strike mission DAN – required by 1500 20 Oct 2020, typical C2 task duration 1 hour, priority 3
2. Control strike mission DAN – required by 1700 20 Oct 2020, typical C2 task duration 4 hours, priority 2
3. Assess strike mission DAN – required by 2100 20 Oct 2020, typical C2 task duration 4 hours, priority 2
4. Plan strike mission DBF – required by 1200 20 Oct 2020, typical C2 task duration 1 hour, priority 3
5. Control strike mission DBF – required by 1400 20 Oct 2020, typical C2 task duration 4 hours, priority 2
6. Assess strike mission DBF – required by 1800 20 Oct 2020, typical C2 task duration 4 hours, priority 2
7. Plan strike mission DDF – required by 1300 20 Oct 2020, typical C2 task duration 1 hour, priority 3
8. Control strike mission DDF – required by 1500 20 Oct 2020, typical C2 task duration 4 hours, priority 2
9. Assess strike mission DDF – required by 1900 20 Oct 2020, typical C2 task duration 4 hours, priority 2
10. Plan strike mission DFN – required by 1700 20 Oct 2020, typical C2 task duration 1 hour, priority 3
11. Control strike mission DFN – required by 1900 20 Oct 2020, typical C2 task duration 4 hours, priority 2
12. Assess strike mission DFN – required by 2300 20 Oct 2020, typical C2 task duration 4 hours, priority 2
13. Plan strike mission DGN – required by 1900 20 Oct 2020, typical C2 task duration 1 hour, priority 3
14. Control strike mission DGN – required by 2100 20 Oct 2020, typical C2 task duration 4 hours, priority 2
15. Assess strike mission DGN – required by 0100 21 Oct 2020, typical C2 task duration 4 hours, priority 2
16. Plan strike mission DJF – required by 1700 20 Oct 2020, typical C2 task duration 1 hour, priority 3
17. Control strike mission DJF – required by 1930 20 Oct 2020, typical C2 task duration 4 hours, priority 2
18. Assess strike mission DJF – required by 2100 20 Oct 2020, typical C2 task duration 4 hours, priority 2
19. Plan strike mission DCF – required by 0300 20 Oct 2020, typical C2 task duration 1 hour, priority 3
20. Control strike mission DCF – required by 0500 20 Oct 2020, typical C2 task duration 4 hours, priority 2
21. Assess strike mission DCF – required by 0900 20 Oct 2020, typical C2 task duration 4 hours, priority 2
22. Plan strike mission DEF – required by 0400 20 Oct 2020, typical C2 task duration 1 hour, priority 3
23. Control strike mission DEF – required by 0600 20 Oct 2020, typical C2 task duration 4 hours, priority 2
24. Assess strike mission DEF – required by 1000 20 Oct 2020, typical C2 task duration 4 hours, priority 2
25. Plan strike mission DHN – required by 0300 20 Oct 2020, typical C2 task duration 1 hour, priority 3
26. Control strike mission DHN – required by 0500 20 Oct 2020, typical C2 task duration 4 hours, priority 2
27. Assess strike mission DHN – required by 0900 20 Oct 2020, typical C2 task duration 4 hours, priority 2
28. Plan strike mission DIF – required by 0930 20 Oct 2020, typical C2 task duration 1 hour, priority 3
29. Control strike mission DIF – required by 1130 20 Oct 2020, typical C2 task duration 4 hours, priority 2
30. Assess strike mission DIF – required by 1530 20 Oct 2020, typical C2 task duration 4 hours, priority 2
31. Plan CAS mission 1 – required by 0200 20 Oct 2020, typical C2 task duration 2 hours, priority 3
32. Control CAS mission 1 – required by 0400 20 Oct 2020, typical C2 task duration 12 hours, priority 2
33. Assess CAS mission 1 – required by 1700 20 Oct 2020, typical C2 task duration 2 hours, priority 2
34. Plan CAS mission 2 – required by 1400 20 Oct 2020, typical C2 task duration 2 hours, priority 3
35. Control CAS mission 2 – required by 1600 20 Oct 2020, typical C2 task duration 12 hours, priority 2
36. Assess CAS mission 2 – required by 0500 21 Oct 2020, typical C2 task duration 2 hours, priority 2
37. Plan DCA mission 1 – required by 0200 20 Oct 2020, typical C2 task duration 2 hours, priority 3
38. Control DCA mission 1 – required by 0400 20 Oct 2020, typical C2 task duration 12 hours, priority 2
39. Assess DCA mission 1 – required by 1700 20 Oct 2020, typical C2 task duration 2 hours, priority 2
40. Plan DCA mission 2 – required by 1400 20 Oct 2020, typical C2 task duration 2 hours, priority 3
41. Control DCA mission 2 – required by 1600 20 Oct 2020, typical C2 task duration 12 hours, priority 2
42. Assess DCA mission 2 – required by 0500 21 Oct 2020, typical C2 task duration 2 hours, priority 2

5 Results

We developed a user interface prototype to show the suggested task allocation. We chose the form of a schedule, which is a familiar paradigm for displaying tasks with temporal extent, but we did not yet have a chance to elicit feedback from the envisioned user community. Separate panels display the allocations to each of the DCNs, listing allocations of capabilities required by the tasks to the resources, including people (Figures 3 and 4). One panel provides a complete view of all the task allocations. The interface lets users view the overall allocations or focus on the allocation and schedule for a single task or for a single capability. We also developed a demonstration control panel to step through the demonstration sequence.

Figures 3 and 4 show the demonstration control window and one of the DCN schedule windows. Clicking "Open Schedules" in the demo control window (Figure 3) without selecting a C2 node from the list opens all of the C2 Node schedule windows.

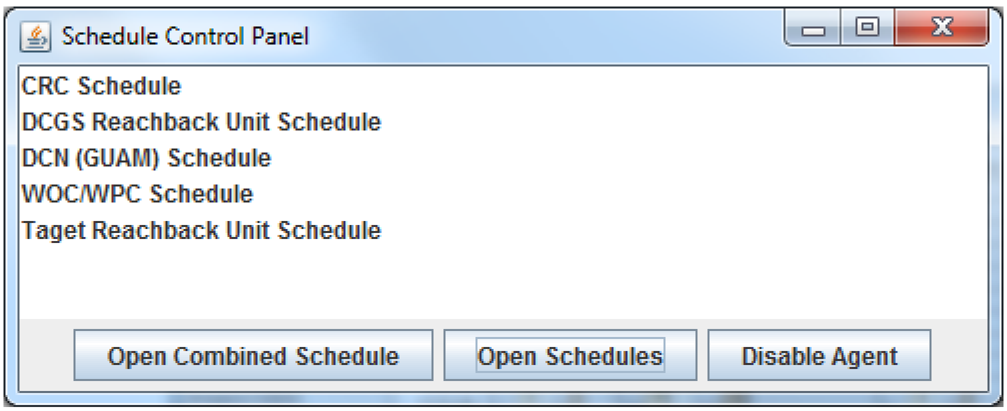

Figure 3. The demonstration control window, where the allocated task schedules of five sample C2 nodes can be viewed. The function of the "Disable Agent" button has not yet been implemented. It would indicate that a DCN is temporarily unavailable. The SPICE system would re-allocate partially completed tasks and tasks that have yet to be started to alternate DCNs.

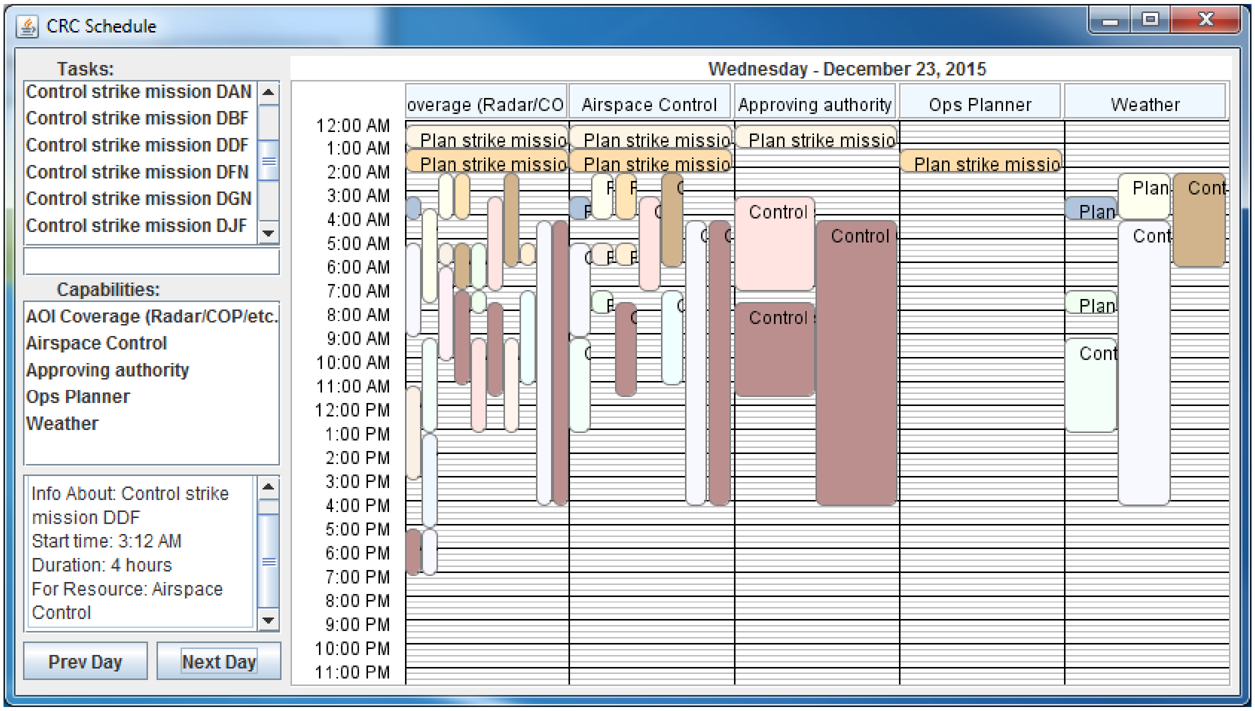

Figure 4. Capability utilization schedule for the CRC node for Day 2 of our two-day scenario. The "Tasks" panel lists all the tasks to which the CRC contributes capabilities. When the user selects one of the tasks, then only allocations for this task are shown. The "Capabilities" panel lists all the capabilities resident in the CRC which are allocated to any of the tasks. These capabilities are also the column headers of the schedule chart. The bottom left panel shows details for a capability allocation selected by clicking on one of the boxes on the schedule chart (capability "Air Space Control" allocated to task "Control strike emission DDF"). The schedule chart shows to which task each capability is allocated at a specific time. Notice that most capabilities can serve multiple tasks, since the tasks only require a portion of the capability. This models capabilities that are not specific to just one task, such as airspace control, and the multi-tasking capabilities of human planners, and the sharing of bandwidth on communications channels.

Figure 5 shows an example of how CLUS-STAR spreads the capabilities required for a single task across multiple C2 nodes, taking advantage of the specialized capabilities available at each DCN. On the other hand, capability requirements for a portion of a capability, e.g., 30% of an ISR Planner, are not split further, but allocated to one of the resources at a single DCN. Since the reward and cost functions are identical for all DCNs and the resource proficiencies are all set to 1.0, the allocation is left up to chance more so than it would with a more realistic scenario configuration.

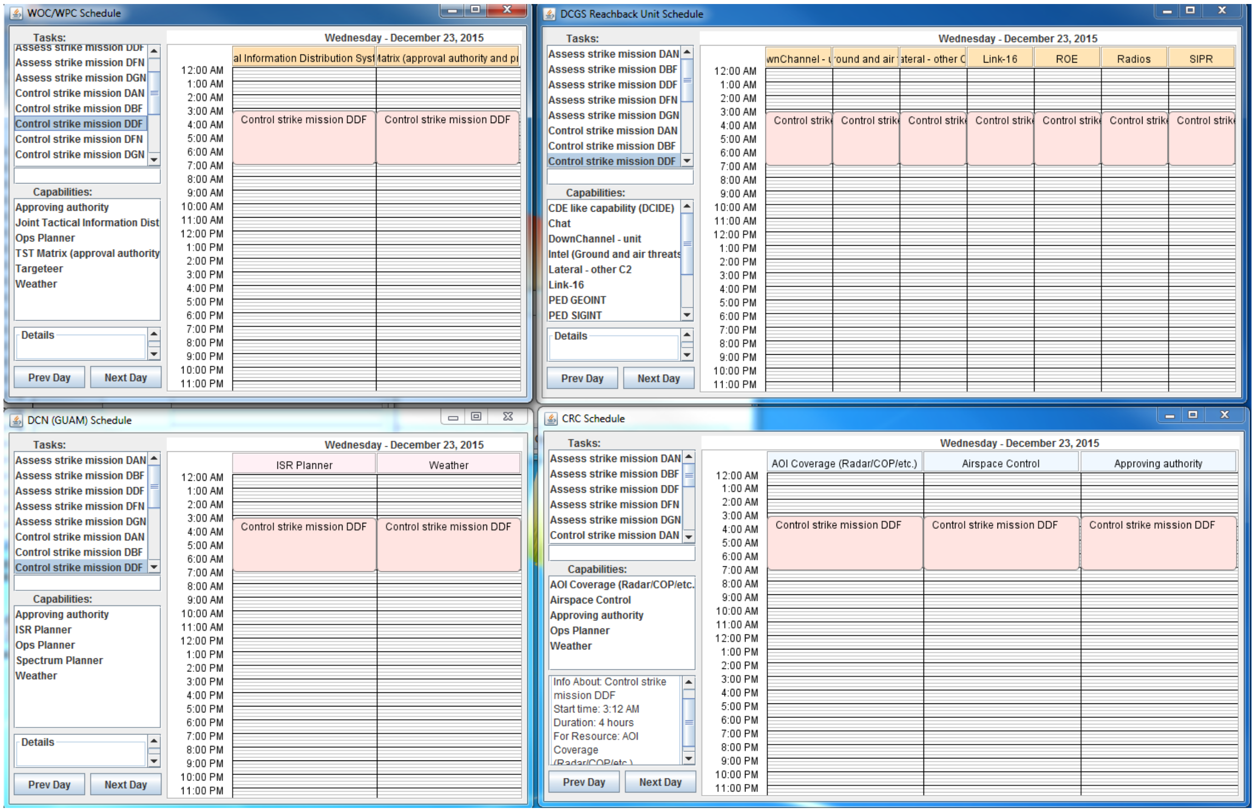

Figure 5. Fourteen capabilities required by the "Control Strike Mission DDF" task are spread across four DCNs. We have selected to display the allocations for only the "Control strike mission DDF" task. For example, the CRC (bottom right) provides AOI (radar) coverage, as it is the node closest to the mission, and also performs airspace control; Guam (bottom left) provides the ISR planner and weather analysis; the WOC (top left) connects to JTIDS; and DCGS Reachback (top right) provides Intel for ground and air threats and ROE evaluation, among other capabilities.

6 Limitations and Future Work

Our own analysis showed that we needed to enhance the specifications of the tasks to include interdependency constraints between multiple capabilities required for the task. Some closely coupled capabilities need to reside in the same C2 node, while others may be distributed across two or more nodes. We have added a capability co-location constraint to our CLUS-STAR algorithm since the time of the experiment.

Our demonstration harness did not yet support triggering the re-allocation of a partially completed task, so that we were not able to experiment with this new algorithm feature.

Future work will focus on two major aspects: (1) validation that the capability will aid the envisioned user community in managing distributed C2 tasks across a changing set of missions, and (2) completion of technical capability.

We plan to validate end user value proposition by gathering SME feedback on the current design using the following steps:

- Increase the realism of the scenario
- Expose the concept to a set of operational users
- Specify a richer set of user interaction beyond inspection of automated results
- Prototype an interactive user interface
- Perform experiments with temporary unavailability of DCNs
    - Activate the functionality of the “Disable Agent” button on the demo control panel
    - Validate the ability to re-allocate a partially completed task

Another extension to CLUS-STAR is suggested by this application: to add a resource requirement constraint that is conditional on a prior allocation decision. For example, a higher level of communications capability between two DCNs will be required if a task is allocated across both these DCNs.

7 Summary

We have successfully completed a first experiment that demonstrates the promise of using an automated tool to suggest efficient and effective allocation of C2 tasks across a number of distributed control nodes with varying staff, systems, and connectivity. We demonstrated the efficacy of an auction-based optimization capability (CLUS-STAR) to optimize the allocations. We suggest future work to extend, validate, and mature this capability to make it relevant to coalition air campaign planning in future A2/AD scenarios.

8 References

(Amador et al., 2014) Sofia Amador, Steven Okamoto, and Roie Zivan. Dynamic multi-agent task allocation with spatial and temporal constraints. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

(Greene & Hofmann, 2006) K. Greene and M. Hofmann (2006), “Coordinating busy agents using a hybrid clustering-auction approach,” In AAAI 2006 Workshop on Auction Mechanisms for Robot Coordination. Boston, MA.

(Guo et al., 2016) Jinhong K. Guo, Alexander Karlovitz, Patrick Jaillet, and Martin O. Hofmann, Probabilistic Auction for Distributed, Robust Resource Optimization (PADRE), publication in preparation.

(Karlsson et al. 2007) Karlsson, Maria, Fredrik Ygge, and Arne Andersson. "Market-Based Approaches to Optimization," Computational Intelligence 23.1 (2007): 92-109.

(Lumezanu et al., 2008) C. Lumezanu, S. Bhola, and M. Astley, Online Optimization for Latency Assignment in Distributed Real-Time Systems, The 28th International Conference on Distributed Computing Systems, 2008 (ICDCS '08), Beijing, June 2008, pp. 752-759.

(Mauadi et al., 2011) M.H.F.B.M. Mauadi, W. Li, and T. Murata, Combinatorial Auction Method for Decentralized Task Assignment of Multiple-Loading Capacity AGV Based on Intelligent Agent Architecture, Second International Conference on Innovations in Bio-inspired Computing and Applications (IBICA), pp. 207-211.

(Moghaddam et al., 2013) M. Moghaddam, J. Davis, and T. Viglas, A Combinatorial Auction Model for Composite Service Selection Based on Preferences and Constraints, IEEE International Conference on Services Compu-ting (SCC), Santa Clara, CA, pp. 81-88.

(Sheu et al., 2010) Ray-Yuan Sheu, Michael Czajkowski, Justin Wetherell, Edwin Simons, Martin Hofmann, Gregory D. Schow, "Real-Time Multi-Agent Adaptive Tactical Pervasive Service-Oriented Collaborations (TAPS)," Invited Paper, Industry Session of ICPS 2010: The 7th International Conference on Pervasive Services, Berlin, Germany, July 2010.

(Zhang et al., 2012) K. Zhang, E.G. Collins Jr., D. Shi, Centralized and Distributed Task Allocation in Multi-Robot Teams via a Stochastic Clustering Auction", ACM Transactions on Autonomous and Adaptive Systems, Vol. 7, No. 2, Article 21, July 2012