14th International Command and Control Research and Technology Symposium
"C2 and Agility"

ENABLING AGILE INFORMATION MANAGEMENT WORKFLOWS FOR C2

Topic: C2 Concepts, Theory, and Policy
Paper 115

Sandeep Mulgund, Ph.D.
The MITRE Corporation
202 Burlington Rd. M/S E015
Bedford, MA 01730
Voice 781-271-6207/Fax 781-271-8875
smulgund@mitre.org

Seth Landsman, Ph.D.
The MITRE Corporation
202 Burlington Rd, M/S M325
Bedford, MA 01730
781-271-7686 / 781-271-2964
landsman@mitre.org

Abstract

Military forces are being called upon to execute a broad spectrum of missions, with a desire to assemble, deploy, and make operational capability within mere days. A critical challenge in enabling such agile capabilities is to ensure that all participants understand their roles, responsibilities, and relationships throughout an evolving mission. From an information management perspective, supporting command and control (C2) systems themselves must be agile and capable of accommodating change. Traditionally, C2 systems attempt to be anticipatory of information needs and provide fixed data model designs intended to support all envisioned use cases. Such designs tend to be brittle and inflexible as mission needs change. We propose an information modeling approach that values run-time flexibility over design-time anticipation, focusing on the essential information that C2 systems must model and share. This paper establishes a functional concept and guiding principles to enable information workflows that support agile C2. We describe and demonstrate a set of principles that can be used to build and share a structured yet flexible representation of essential C2 information that can support the evolution of agile C2 capabilities.

Keywords: C2 architecture, agile C2, data modeling

1. Introduction

Military forces are being called upon to execute a broad spectrum of missions, ranging from tradition force on force, to counter insurgency, to precision strike, to disaster relief and others. The diversity of these missions and desire to assemble, deploy, and make operational capability within mere days places demands on the commanders executing C2, C2 processes, and C2 systems. A critical challenge in such an agile system is ensuring that all participants understand their roles, responsibilities, and relationships to peers, subordinates, and commanding C2 participants throughout an evolving mission. Importantly, each echelon or element must have a clear understanding of their role in the context of achieving the overall mission intent, often called “commander’s intent”. As intent evolves into action, it is also critical to ensure that all cooperating participants have a consistent view of the evolving operational environment.

The North Atlantic Treaty Organization (NATO) has defined a set of C2 maturity levels that map to an overall level of net-enabled capability (NEC) maturity level. The model is defined in the context of a coalition force consisting of some number of civilian and military elements (inter-agency or whole-of-government) from the various NATO nations (Moffat, 2008). These levels define a progressively increasing scale of C2 maturity and agility, and in theory they can be applied to individuals and organizations of any size. Moving up this scale calls for a progressively increasing ability to establish shared intent and awareness, mutual roles, and to execute coordinated, synchronized, and mutually reinforcing actions. Realizing the highest level of C2 maturity defined by the model, *agile C2*, poses a considerable challenge for the community to address, depending on a combination of organizational principles, training, and technology. In the technology space, numerous efforts are now under way for net-centric data exposure and exploitation, including the development of core C2 vocabularies, service oriented architectures, and decision support tools such as user defined operational pictures. However, what is less clear is how to define, organize, and manage the composite information products needed to enable agile C2 through the course of a mission thread that may consist of the following elements:

- Day-to-day operations, taking place in the context of general strategic goals and mutually understood intent.
- Detection of an event of concern that may trigger more focused activity to respond to the event.
- Assessment of the situation.
- Definition and selection of a preferred course of action consistent with overall commander's intent.
- Mission execution and assessment in relation to established intent.

The information products needed to support agile C2 exist at multiple C2 echelons, ranging from tactical up to national/strategic. Their content becomes more focused as events move towards mission execution, and consists of a mixture of live data feeds, human-generated content such as plans and goals, and products that are derived from original source material (for example, assessments attached to raw sensor data). Many different tools will be used by different users to construct the overall picture needed to enable agile C2. Managing the information products created with these tools requires definition of *workflows* that make it possible to synthesize the overall information ensemble that is needed by cooperating entities. We believe that realizing the vision of agile C2 as defined by the NATO model requires a strategy for defining and managing these information products in such a way that all participants have a complete and unambiguous shared understanding of the intent, objectives, plans, and current status throughout a mission. This paper presents a methodology for realizing these requirements. Defining and enabling these workflows is a critical next step in the evolution of net-centric warfighting capabilities.

Section 2 of this paper provides an overview of essential information requirements for C2, followed by a discussion on C2 agility in section 3. Based on these requirements, section 4 then presents a strategy for how to model complex information products for C2 in an agile, enterprise-scalable manner. Section 5 applies these principles to an example for crisis action planning. Finally, section 6 summarizes this paper's findings.

2. Information Requirements for C2

The term command and control is quite broad in its scope, ranging from broad national/strategic levels down to very narrow tactical levels. The nature of the task to be accomplished will drive the nature of the resources involved. This can range from something that can be accomplished with assets organic to an organization to something requiring the assembly of a large heterogeneous coalition with diverse resources. Alberts and Hayes (2006) identify the following functions as being associated with the C2 of a given undertaking:

1. **Establishing intent**: What is the mission to be accomplished? Without some purpose, the notion of command and control makes little sense. This expression of intent exists in the context of some situation and the entities that will fulfill the mission.
2. **Determining roles, responsibilities, relationships, and constraints**. The term C2 implies the existence of multiple entities. Typically, different entities play different roles.

Establishing roles and responsibilities serves to enable, encourage, and constrain specific types of behaviors. How is this information codified and conveyed through C2 networks?

3. **Monitoring and assessing the situation and progress**. As events progress in the battlespace, decision-makers must be able to monitor activities and adjust accordingly. Making these assessments depends on being able to detect relevant elements of information, interpreting their meaning, and projecting likely future states and mission impacts. Adjustments may manifest as changes in intent or plans, modifications of roles and responsibilities, or changes in rules and constraints of operation. This relates to the topic of maintaining *situation awareness*, which has been explored at tremendous length in the literature (e.g., Endsley, Bolté, & Jones, 2003).
4. **Inspiring, motivating, and engendering trust**. These three functions relate to *leadership*, and determine the extent to which individual participants are willing to contribute and the nature of interactions that take place.
5. **Training and education**. A force's competence and professionalism will have profound impact on their ability to execute a mission. For a current operation, levels of training and education are a given. In the short run they can be affected modestly, but in the long run they can be transformative. The effectiveness of any new capability will depend on suitable education and training methods.
6. **Provisioning resources**. Adequate quantity and type of resources are critical to the success of any endeavor. Provisioning occurs from both an enterprise and mission perspective, as well as from a long-term and short-term perspective. Adequate provisioning depends on the existence of the right types of resources, knowledge of their availability, and also on the feasibility of delivering them where they are needed in the requisite timeframe. This may be more of a challenge in "immature" theaters lacking adequate port facilities, airfields, ground transportation, etc. (Joint Staff, 2000).

From an information management perspective focused on conducting operations, it is items 1, 2, 3 and 6 that are of greatest interest here. They relate to the dynamic information needs that are most associated with "real time" C2 through the course of an operation. Numerous techniques exist for enterprise data modeling, but from a conceptual perspective, what is the right organizing principle for structuring that knowledge? We believe that that the topic of *intent* can provide that organizing framework, as it guides what it is that military forces must accomplish. Builder *et al* (1999) provide an analysis of the success/failure of numerous historical military campaigns from the perspective of commander's intent, highlighting its importance as a basis for effective C2. They define a "command concept" that focuses on the essence of C2 – command intent and ideas about how to satisfy that intent. A command concept is a capturing of the intellectual functions of command and can be defined as a *vision* of a prospective military operation that informs the making of command decisions during that operation. Among other attributes, this vision can include:

- **Time scales** of operation that reveal adequate preparation and readiness, not just of the concept but of the forces tasked with its execution.
- **Awareness of the key features** of the battlespace that will enable realization of the concept (weather, terrain, etc.). Maintaining this awareness can be especially challenging if the battlespace spans multiple physical areas of responsibility (AORs).
- A **force structure** consistent with the tasks to be accomplished.

- **Congruence of the concept** with the means of executing it.
- **Intelligence** about an adversary's objectives, including confirming and refuting signs to look for throughout the engagement. This includes what the enemy is trying to accomplish, not just what his capabilities and dispositions might be.
- **What forces should be able to do** and how to do it. This includes required deployments, logistics, and schedules, as well as the nature of expected clashes and what might be expected in the confusion of battle.
- **Indicators of failure or flaws** in the command concept, and ways of identifying or communicating information that would change or nullify it.
- A **contingency plan** in the event of failure.

Examining the two lists, we develop a sense of the information ensemble in which C2 systems must be conversant. Builder *et al* argue that the critical purpose of C2 systems should be to provide decision-makers with information needed to develop and refine command concepts, to communicate them across the chain of command, and to integrate information that indicates success or failure as a mission progresses. Accordingly, C2 systems must be designed to provide high fidelity representations of information pertinent to the command concept. While their focus is on traditional command hierarchies, the essential underlying principle -- the importance of making explicit what is to be done and how to do it – is equally if not more applicable to enabling more agile C2 where participants are distributed and heterogeneous in nature.

Our hypothesis is that an effective information management strategy for agile C2 is to organize the constituent elements of information in a manner consistent with the command concept, and then lay in assessments relative to that vision as events progress. Such a shared repository of knowledge provides a continuously evolving "data picture" that expresses:

- What are distributed forces attempting to accomplish?
- How will they do it?
- What is the assessed progress against plan, and are any changes necessary?

While such information might be depicted visually through a tool such as a user defined operational picture (Mulgund & Landsman, 2007), our interest here is in defining how to model and build such composite data products for agile C2. In the next section we characterize agility in related information management processes for C2, leading to a discussion of how to model and use these data products.

3. Characterizing C2 Agility

NATO has defined a set of C2 maturity levels that map to an overall level of net-enabled capability (NEC) maturity level. The model is defined in the context of a coalition force consisting of some number of civilian and military elements (inter-agency or whole-of-government) from the various NATO nations (Moffat, 2008). These levels define a progressively increasing scale of C2 maturity and agility, and in theory they can be applied to individuals and organizations of any size. Realizing the highest level of C2 maturity defined by the model, *agile C2*, poses a considerable challenge for the community to address. The NATO Network Enabled

C2 (N2C2) Maturity Model defines five levels that differ from one other meaningfully in terms of the applied C2 approach. C2 objectives at each level are defined as:

- **Conflicted C2**: The only C2 is that exercised by individual contributors over their own forces or sub-elements. At this level, no C2 is being exercised at a collective level, and each entity pursues its own intent with independent actions.
- **Deconflicted C2**: The avoidance of adverse cross-impacts among participants by partitioning the problem space.
- **Coordinated C2**: Increasing overall effectiveness or efficiency by: (a) seeking mutual support for intent; (b) developing relationships and linkages between and among entity actions to reinforce or enhance effects; (c) pooling resources to accomplish results that are not otherwise possible; and (d) increasing sharing in the information domain to increase quality of information.
- **Collaborative C2**: Developing significant synergies by: (a) negotiating and establishing shared intent; (b) establishing or reconfiguring roles; (c) coupling actions; (d) sharing non-organic resources; (e) pooling organic resources; and (f) increasing interactions in the cognitive domain to increased shared awareness.
- **Agile C2**: Providing an endeavor with additional C2 approach options that involve entities working more closely together and with the ability to identify and implement the most appropriate C2 approach given the situation (e.g., mission, operating environment, and set of coalition partners and contributing entities). This level of C2 maturity depends upon achieving a high degree of shared understanding of a common intent. It requires a rich and continuous set of interactions among participants, involving widespread information-sharing to allow the build-up of trust, shared understanding, and the willingness and ability to self-synchronize.

Moving up this scale calls for a progressively increasing ability to establish shared intent and awareness, mutual roles, and to execute coordinated, synchronized, and mutually reinforcing actions. Our interest here is in establishing the essential characteristics of the information infrastructure needed to support progressively increasing agility. We focus not on implementation approaches such as service oriented architectures, but rather on the strategy for modeling and building the composite information products needed to stitch together a structured, scalable, evolving picture of an operation.

4. Building Composite Information Products for Agile C2

Existing guidance such as the Department of Defense’s Net-Centric Data Strategy (NCDS) (DoD, 2003) discusses how to manage information in a net-centric environment. It focuses on key tenets of making data visible, accessible, understandable, trustable, and interoperable. However, it does not address to create scalable, flexible enterprise-level data models. Each participant in a collaborative endeavor must understand its objectives and constraints, and how its mission relates to others ongoing. As events progress and original plans change, it is necessary for decision-makers and executing forces to understand the impacts. The challenge lies in how to make available that information ensemble in such a way that different participants can access what they need, and share any products that they develop (details plans, execution assessments, status updates, etc.). How can we define the *business object* (i.e., the representation of the information in the domain of C2 in which systems much be conversant) that encodes the command concept in such a way that it lends itself to the requirements of agile C2? We believe that this calls for a structural representation that lays out relationships between different elements of information, modeled in such a way it is adaptable to change through the course of a mission.

This section defines a set of strategies for creating information models of the command concept in such a way as to support the requirements of agile C2. Section 4.1 lays out a set of essential requirements for this modeling challenge. Section 4.2 presents three concrete modeling approaches for satisfying these requirements. Finally, section 4.3 then shows how to realize what we believe to be the most promising approach using existing technologies and techniques.

4.1 Essential Requirements

Our objective is to identify a strategy for modeling a business object that provides a flexible, physical representation of the command concept, with each information element representing a distinct component thereof. To achieve the goal of agile C2, we require information representations that are flexible in terms of their content, structure, and ability to accommodate change. A system that makes rigid assumptions about the information desired by its users will constrain information sharing and limit the degree of C2 agility possible to what was conceived at design time. As a situation unfolds, unanticipated events will make new demands on C2 systems in terms of the types of information needed, requiring that they be adaptable to such requirements. We believe that a suitable business object model that is sufficiently flexible to support the needs of agile C2 has three essential requirements:

- It must support changing the content and structure of the business object, as events evolve and new sources of information become important, new areas of focus develop, and dynamic relationships form between participating entities
- It must be transparent to change, so that systems and users who are relying on it can detect and understand the impacts of those changes
- It must afford different perspectives and views on the underlying data, so that each participant in a collaborative endeavor can access and use what they need for their mission, without having to be overburdened with unrelated elements

First, information within the business object will be subject to change as a mission progresses. For example, consider a search and rescue mission where different participants may work together collaboratively. It may be the case that Coast Guard and Navy forces may perform search and rescue at sea, while National Guard and local law enforcement may be responsible for land-based operations. Information products of interest might include specific areas of responsibility, asset descriptions (helicopters, ships, etc.), and incident reports provided by each executing entity. What is needed is a way of blending these information products together to provide an integrated picture of the evolving mission, defined in such a way that the specific information contents are not constrained to what was conceived when supporting systems were designed. To do so requires agility in incorporating new information products.

Second, over time information elements related to the command concept will change, reflecting mission progress, updated intelligence, refined plans, and other factors. Each participant will need to receive updates to that their view maintains synchronization with the overall command vision. However, broadcasting every change as it happens may not be effective. Different participants may only be interested in a subset of the overall picture that is relevant to their mission. Network limitations may impose other constraints. For example, a forward stationed unit may have a poor network link and want to prioritize receipt of only that information that is most directly relevant to his tasking. A mission planner, on the other hand, may require a more global picture of the current situation to make timely updates to evolving plans. In either case, when data changes, the nature of those changes should be transparent and easy to understand for consumers, to that they can incorporate them into their ongoing business processes and understanding of the operational environment.

Finally, different users and systems in a collaborative endeavor are likely to have different information needs through the course of a mission. A mission planner and a intelligence officer may be following the same event and have similar goals, but the key information elements they use, and how those information elements fit into the larger command concept may differ. In this case, a mission planner wants to see the list of missions, and how a mission, forces, and targets relate. An intelligence officer wants to see intelligence reports, and their relationship to other information elements. Each consumer may thus want to orient the picture in accordance with their own perspective, while maintaining consistency with the global picture.

4.2 Approaches for Modeling Composite Business Objects

The structure, content, and how the business object is made available to distributed participants all have an impact on its flexibility of use. We have identified three modeling approaches, all of which have different tradeoffs. The three approaches are shown in Figure 1 and are to (from left to right):

- Define a single monolithic object that contains the entire data ensemble of interest in a top-down hierarchy
- Define multiple objects that map to each major key C2 function, such as intelligence, logistics, planning, etc.
- Define a *graph* of fine-grained information elements, organized according to the requesting consumer's needs

The first approach defines a single business object that models the entire command concept from constituent elements such as intelligence products, commander's intent, human intelligence (HUMINT), signals intelligence (SIGINT), plans, etc. In this approach, each consumer gets the same object, and all changes to that object are broadcast to all consumers. The object itself is fixed in terms of structure and content based on design-time decisions. This approach has the advantage of being very straightforward to implement using technologies such as extensible markup language (XML) schemas. However, it is limited in that the topology of the model is typically fixed at design time, and difficult to change thereafter. Since the entire domain is captured in a single object, detecting fine-grained changes within any of its constituent components can be a challenge. In particular, if just one small item out of the monolithic object changes, what should interested participants receive? If they receive the entire object, they must traverse it to understand what changed, and then take any necessary actions. Broadcasting the entire object upon every change can also place a large burden on communications networks. If they receive just a fragment of the object, it requires them to know how to incorporate that fragment into their overall picture of the command concept. It also only affords one perspective on the data, based on the top-down hierarchy established at design time.

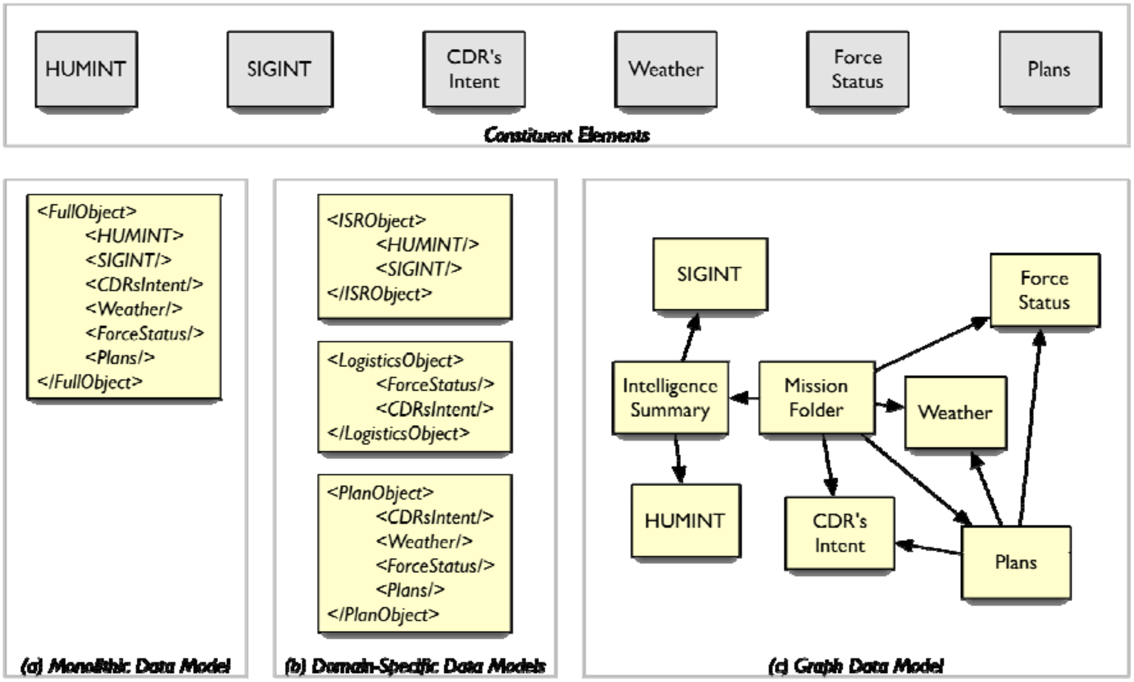


Figure 1. Alternative Data Modeling Approaches

To provide more flexibility and fine-grained resolution of data products of interest, Davies *et al* (2008) propose defining multiple business objects for each major domain of information in the enterprise. In this case a consumer chooses which business object(s) are relevant for its mission, and interacts with services that provide those particular views of the data. Unlike the previous case, there is now finer granularity of the data that comprises the overall picture. A new business object may be added without affecting existing consumers of models that have been established previously. However, defining a finite set of combinations and permutations of the command concept's components may be a challenge, and it shares the limitation of the preceding approach that extending any of these models in a predictable way at run-time is difficult.

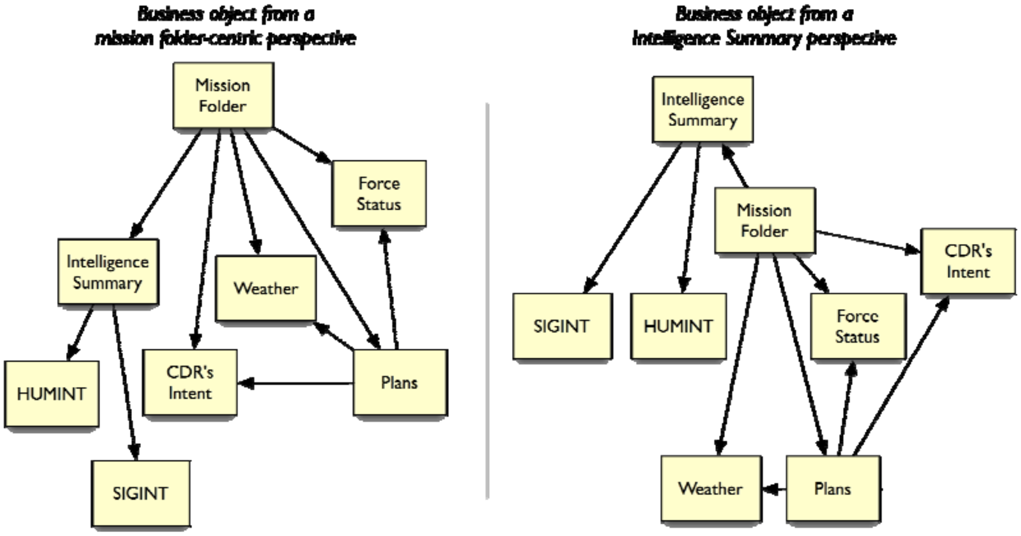
In both of these cases, the business object is defined as a fixed data structure. Traditionally, such a synthesized object would be created as an XML or other hierarchical representation. However, how is the data represented when it does not neatly fit into a hierarchy that can be specified at design time? A command concept is an interrelated *web* of knowledge, which contains interdependencies that lead to cycles and multiple references between information elements. For example, both a plan and a force status summary may contain a reference to a specific unit that will execute some task. To express such a relationship in a traditional XML hierarchy, the information element describing the unit would most likely be copied in multiple locations or require using complex rules for connecting one part of the hierarchy to another. Either approach requires force-fitting the data into an XML tree hierarchy. Representing such a structure as a hierarchical XML document can result in a loss of fidelity or difficulty in discerning precise relationships.

The third approach is to express the data of interest as a *directed graph*, which consists of a set of *nodes* each encoding some element of information, connected to one another through *edges* or *links* that define specific relationships between the nodes. A graph model eliminates the artifice of a forced top-down hierarchy, and allows for expressing complex relationships among nodes through the links that connect them. In the example shown in Figure 1c, both the **mission folder** node (an organizing hub for this small graph) and the **plans** node link to the **weather** node, a relationship that would be difficult to model easily in a top-down hierarchy. Some nodes in the graph contain many links (e.g., **mission folder** or **plans**) and act as hubs for associated information, while others contain just one link (e.g., the **HUMINT** node, referred to only by the **intelligence summary** node). If the nodes in the network are the essential nouns in the domain, the links are the verbs that connect them (e.g., sortie *executes* plan, plan *defines* objective).

Recent research has shown that such a graph model has proven to be a highly recurrent pattern, found in domains such as diverse as transportation networks, connectivity of web pages on the world wide web, and disease propagation modeling (Barabasi, 2003, 2007). Barabasi has shown that rather than having a random pattern of connectivity, such networks tend to have a small set of highly connected nodes, with progressively larger numbers of less-connected nodes. He introduces the term *scale free network* to describe networks following a power law distribution in their connectivity patterns, where the fraction $P(k)$ of nodes in the network having k connections to other nodes scales as $P(k) \sim k^{-\gamma}$. The constant γ is typically in the range of 2-3. In a random network there is a characteristic scale to the network, meaning that most nodes have the same number of links, typically in a Gaussian distribution about the mean. However, in a scale free network there is no single node that is characteristic of all others, and the network has no intrinsic scale constraining inter-node relationships or the network's size. Barabasi found that as such networks evolve, they do so not in a random manner but in a way that exhibits *preferential attachment*, with nodes that are highly connected becoming even more so as the network grows. Scale free networks have the property of being resilient to change, meaning that adding or removing nodes from the network has little impact on the rest of the network.

Given a scale-free network's ability to expand over time in a way that forms clusters around highly connected nodes, we believe that it provides a robust conceptual framework for the design of a composite business object that encapsulates the web of information in a command concept. Rather than forcing the information ensemble into an artificial hierarchy, we can express all the linkages in the network in such a way that promote deep understanding of the relationships between disparate pieces of information. Adding in new information elements or new hub nodes will not affect the existing properties of a composite object modeled in this way. The result is that the business object can change and evolve during course of a mission to reflect new plans, sources of information, or other inputs not anticipated at design time. We believe that this property of extensibility in response to unanticipated representation requirements is a key enabler for agile C2, since it provides for building complex, extensible webs of information that are not constrained *a priori*. Yet, the structure of nodes and links encodes the relationships and constraints that will be of interest to consumers of the data. Such a fine-grained, graph-based data model readily supports change detection and interpretation. The atomic unit in the graph is each individual node, and when a node's content or relationships with other nodes change, messages that encode those updates can be published onto the C2 network. A recipient of such a message can establish which node has changed, and via that node's linkages understand the global impact.

Finally, graph-based models lend themselves to being viewed from different perspectives. Figure 2 shows two different graph representations of the same collection of information, though each graph uses a different node as its hub to organize the rest of the content. The image on the left centers on the **mission folder**, suited for a decision-maker requiring a global view of the mission. The image on the right centers on the **intelligence summary**, suited for an intelligence analyst who may be interested primarily in incoming intelligence products. Although both graphs contain the same information, accessing it from different hub nodes eases access to different information elements, while preserving the accuracy of relationships with the rest of the overall ensemble. Both users might receive exactly the same graph, but use different nodes as the hub for navigating around it. If so desired, the intelligence analyst might choose to receive only the intelligence summary and its immediate children (**SIGINT** and **HUMINT**), if the other elements of information are not of interest.


Business object from a mission folder-centric perspective

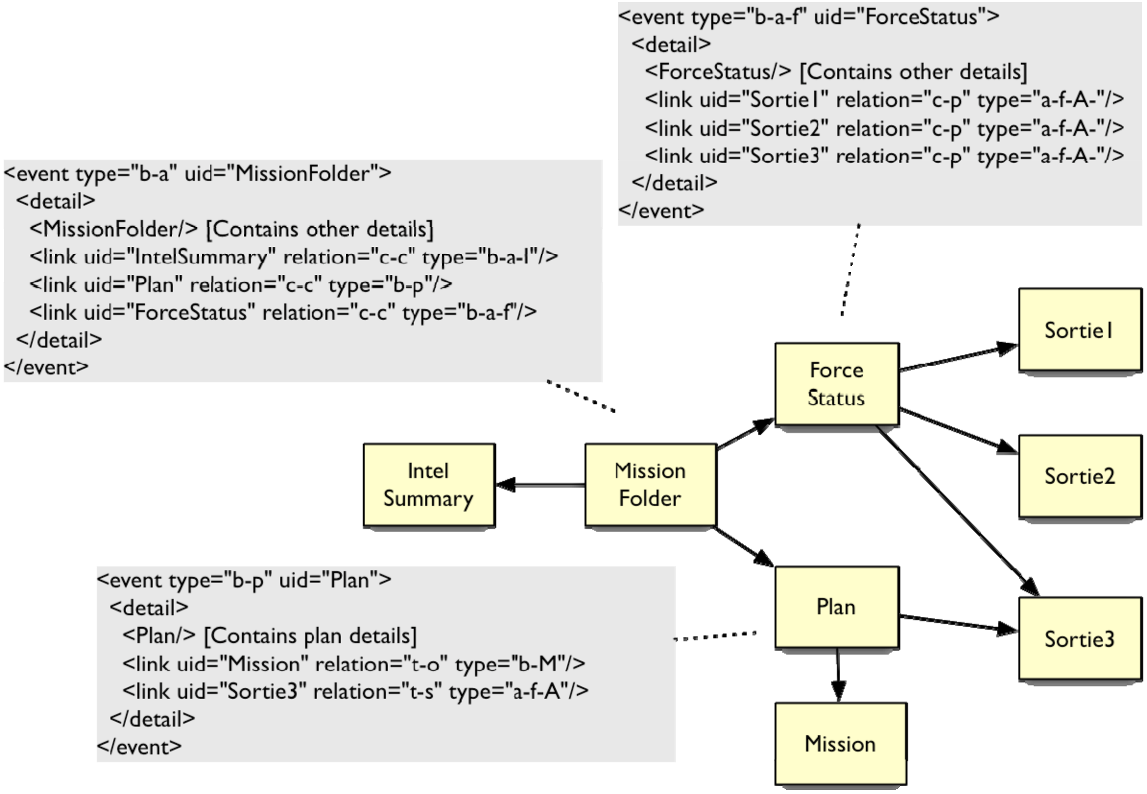
Business object from a Intelligence Summary perspective

Figure 2. Two perspectives on the same business object

4.3 Realizing a Graph-Based Business Object

The graph model consists of a collection of nodes and links. Each node encodes some fine-grained information element, and the links describe relationships with other nodes. How do we encode such a model using available tools and techniques, and make them available over a network in such a way that is compatible with NCDS tenets? One existing technology that can be used to model such graphs is *Cursor On Target* (CoT) (Miller, 2004). CoT is a machine-to-machine language designed to communicate critical what/where/when battlefield information in a concise data structure. It has been used successfully in a number of live operations and experimental settings. It provides a mechanism for defining whole/part linked relationships in accordance with the preceding discussion.

Figure 3 presents a sample business object and its CoT representations. An information element represented as a CoT data structure is an XML document that contains a set of core identifying information, such as the element's unique identifier, type specifications, and other information such a reporting time, location, and other details. In the sample shown, a mission folder is defined as a "b-a" object, meaning it is metadata (**b**its) about an **a**ggregated set of other information. Force status is shown as having type "b-a-f", defining it as **b**its about an **a**ggregated set of reports about **f**riendly units.



```
<event type="b-a-f" uid="ForceStatus">
  <detail>
    <ForceStatus/> [Contains other details]
    <link uid="Sortie1" relation="c-p" type="a-f-A-"/>
    <link uid="Sortie2" relation="c-p" type="a-f-A-"/>
    <link uid="Sortie3" relation="c-p" type="a-f-A-"/>
  </detail>
</event>
```

```
<event type="b-a" uid="MissionFolder">
  <detail>
    <MissionFolder/> [Contains other details]
    <link uid="IntelSummary" relation="c-c" type="b-a-I"/>
    <link uid="Plan" relation="c-c" type="b-p"/>
    <link uid="ForceStatus" relation="c-c" type="b-a-f"/>
  </detail>
</event>
```

```
<event type="b-p" uid="Plan">
  <detail>
    <Plan/> [Contains plan details]
    <link uid="Mission" relation="t-o" type="b-M"/>
    <link uid="Sortie3" relation="t-s" type="a-f-A"/>
  </detail>
</event>
```

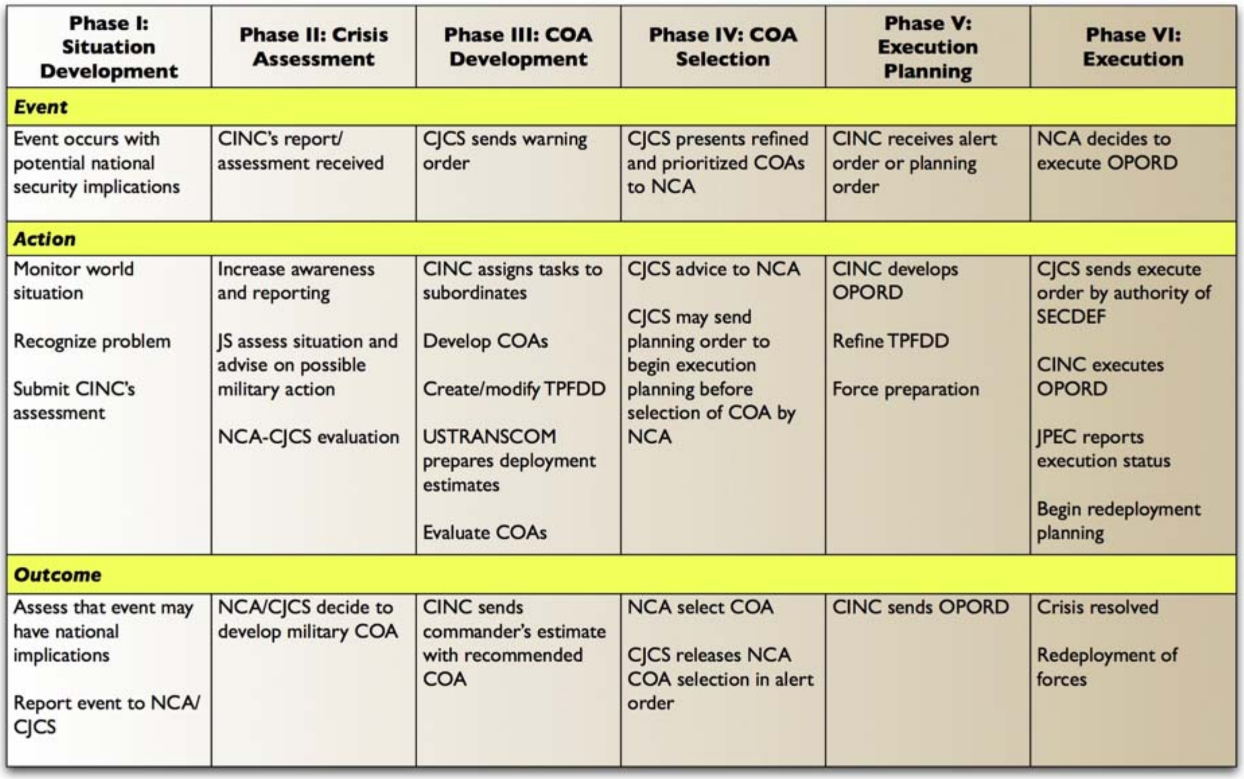
Figure 3. Example graph modeled using Cursor on Target

Each of the CoT documents each contain information about relationships with other nodes, defined by the <link> elements shown in the illustration. The link element contains three pieces of information: the object to which a link is being defined (specified by its unique identifier), the type of object that is the target of the link, and the type of relationship specified. The relationships in the graph shown differ depending on the originator and target of the link. A mission folder is a container of many other correlated information elements, so it forms a correlated child (c-c) link with the other elements in this graph. The **Force Status** node is a composite of individual blue force reports, and it forms a **c**omposite **p**arent (c-p) relationship with **Sortie1**, **Sortie2**, and **Sortie3**. The plan connects a mission and an assigned sortie (i.e., commitment of resources), so it forms a **t**asking **o**bject (t-o) and **t**asking **s**ubject (t-s) relationship with the two elements, respectively.

The graph in Figure 3 makes clear the limitations of a traditional hierarchical business object. **Force Status** aggregates reports about the status of **Sortie1**, **Sortie2**, and **Sortie3**. **Sortie3** is also the subject of **Plan**. In a traditional tree-style hierarchy, both information elements could not have a directed relationship with **Sortie3**, even though such a relationship clearly can exist in the physical world. The graph model preserves that relationship, and techniques such as link analysis can be used to discern relationships between nodes. For example, **Plan** and **Force Status** are both components of **Mission Folder**. However, they are also related in that **Force Status** contains information about **Sortie3**, which is a resource tasked by **Plan**. Understanding such relationships will be essential for agile C2, in which cooperating entities need a detailed understanding of shared intent, respective roles and relationships, constraints, and status. The graph model provides a means of encoding such information precisely.

5. Example Workflow: Crisis Action Planning

To illustrate the preceding concepts in action and to demonstrate how we can evolve a graph model representation of a command concept over time, we now describe how they would apply to a specific type of mission. The U.S. Department of Defense has defined a set of procedures for crisis action planning (CAP) that it uses to respond to events that arise with relatively little or no advance warning. CAP begins with the occurrence of an event somewhere in the world with potential national security implications, and it ends with a resolution to that crisis via military or other means. Figure 4 illustrates the CAP process flow.



Phase I: Situation Development	Phase II: Crisis Assessment	Phase III: COA Development	Phase IV: COA Selection	Phase V: Execution Planning	Phase VI: Execution
Event					
Event occurs with potential national security implications	CINC's report/ assessment received	CJCS sends warning order	CJCS presents refined and prioritized COAs to NCA	CINC receives alert order or planning order	NCA decides to execute OPORD
Action					
Monitor world situation Recognize problem Submit CINC's assessment	Increase awareness and reporting JS assess situation and advise on possible military action NCA-CJCS evaluation	CINC assigns tasks to subordinates Develop COAs Create/modify TPFDD USTRANSCOM prepares deployment estimates Evaluate COAs	CJCS advice to NCA CJCS may send planning order to begin execution planning before selection of COA by NCA	CINC develops OPORD Refine TPFDD Force preparation	CJCS sends execute order by authority of SECDEF CINC executes OPORD JPEC reports execution status Begin redeployment planning
Outcome					
Assess that event may have national implications Report event to NCA/ CJCS	NCA/CJCS decide to develop military COA	CINC sends commander's estimate with recommended COA	NCA select COA CJCS releases NCA COA selection in alert order	CINC sends OPORD	Crisis resolved Redeployment of forces

Figure 4: Crisis Action Planning Flow. From Joint Staff (2000).

The situations for which CAP applies are dynamic, with the body of knowledge used to drive assessments and planning growing hourly with latest information updates and intelligence reports (Joint Staff, 2000). Procedures begin when the situation develops. In Phase I, a theater commander recognizes a potential national security significance of the event and reports it along with related assessments to the national command authority (NCA, now often referred to as *national leadership*) and the Chairman of the Joint Chiefs of Staff (CJCS). During Phase II, national leadership assesses the diplomatic, economic, and informational implications and may decide that a possible military response be prepared. In Phase III a designated commander then develops potential military courses of action (COAs) to respond to the situation, formulated as a part of the Commander's Estimate. Should national leadership decide to use military forces to resolve the crisis, it will select a COA for full development into a detailed plan in Phase IV. Detailed planning then occurs in Phase V, with the designated commander preparing a detailed operations order (OPORD) to execute the selected COA. Based on direction from national leadership, mission execution then occurs in Phase VI. A key portion of the execution phase is an assessment of the outcome, which may drive further action or force redeployment.

While this description presents the process as sequential, many steps can occur concurrently, or be skipped altogether provided that no critical factors are overlooked. Exact procedural flows depend on the time available to complete planning and the nature of the crisis. This process flow provides a rich example for demonstrating how to apply the concepts presented in this paper for defining information workflows to support agile C2. In particular, the overall ensemble of information related to this event grows over time, subject to refinement as the understanding of what is happening improves and general response COAs turn into concrete detailed plans.

Figure 5 illustrates notional information products that can come into play through the course of crisis action planning. Today, many of these artifacts exist in disconnected form from source data and from each other (e.g., Powerpoint briefings documents, information displays on specific systems, screenshots, etc.). Artifacts such as briefings or other documents are disconnected from what may be going on in the real world, limiting their utility as circumstances change. Different participants may provide different pieces of information, deriving from a combination of machine-to-machine data (e.g., threat force locations, blue force status, and weather) and human-generated content (assessments, tasking, or assumptions). In the envisioned workflow, all of these information elements would be "glued" together via the organizing construct of the mission folder, which provides a structured, flexible representation of the command concept and the evolving battlefield information that relates to it.

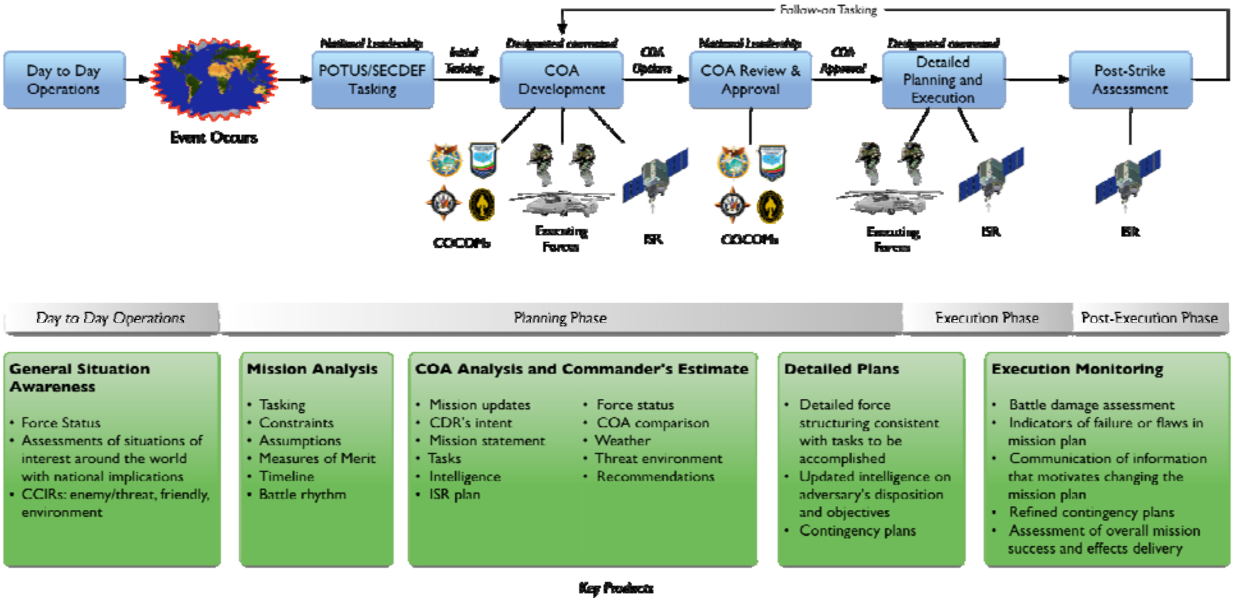


Figure 5: Example Information Products through Crisis Action Planning Mission

The figures that follow illustrate how the connected graph of information products that support the operation would evolve. Their contents are simplified in the interest of readability. Information products evolve and grow in accordance with the figure above, namely:

1. **General situation awareness**: Monitoring events of interest around the world, and maintaining awareness of force status and the commander's critical information requirements (CCIRs).
2. **Mission analysis**: After detection of an event of interest, analysis of the potential mission.
3. **Commander's estimate development**: Definition of mission intent, and development of alternative courses of action. In essence, preliminary definition of the command concept.
4. **Detailed planning**: Once a COA is selected for execution, detailed planning occurs to flesh out the command concept.
5. **Execution monitoring**: The final step is mission execution and assessment of its outcome, which may lead to subsequent tasking and/or force redeployment.

Our objective here is to depict how a graph structure of knowledge can grow over time to encapsulate the command concept, refined in an agile way as events progress and new information becomes known. Figure 6 shows the initial conditions, maintenance of a global picture organized by the CCIRs. CCIRs establish a comprehensive set of information requirements that the commander has identified as being critical to facilitate timely decision-making. The dramatically simplified representation shown here consists of priority intelligence requirements (PIRs), characteristics of the area of responsibility (AOR), and force status information. In turn, the PIRs may consist of size, activity, location, unit, time, and equipment (SALUTE) reports and other intelligence summaries. AOR information includes weather and terrain details. The critical observation here is that the CCIRs provide an organizing construct for navigating this global picture, which users may access and exploit through a variety of mission tools.

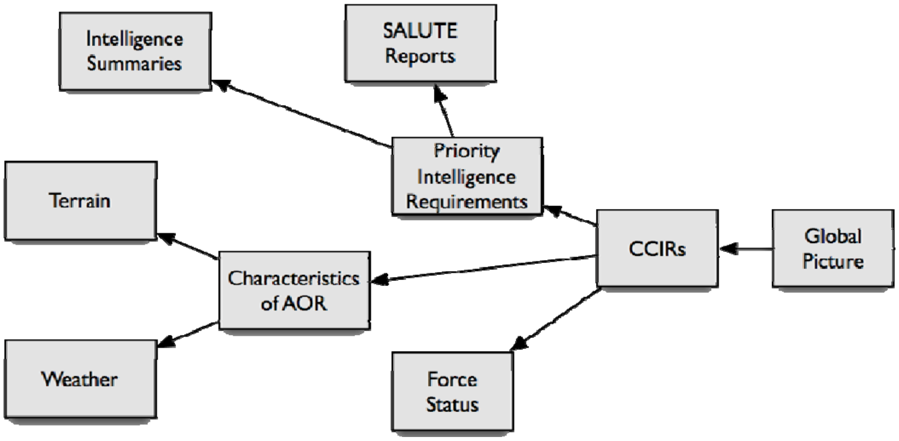


Figure 6: Step 1 – Global Picture Organized by Commander's Critical Information Requirements

Upon detection of the crisis event of concern, mission analysis occurs to assess the crisis and establish whether a response is warranted. In accordance with the workflow described in the preceding section, a mission folder is defined to tie together the relevant pieces of information that will guide decision-making and execution of any response. As Figure 7 illustrates, it links to the CCIRs and also to a mission analysis product. In turn, the mission analysis also relies on the CCIRs as a basis for its contents. The subordinate structure of the mission analysis is not shown for brevity. The mission folder now becomes the hub for all related information products for this crisis event.

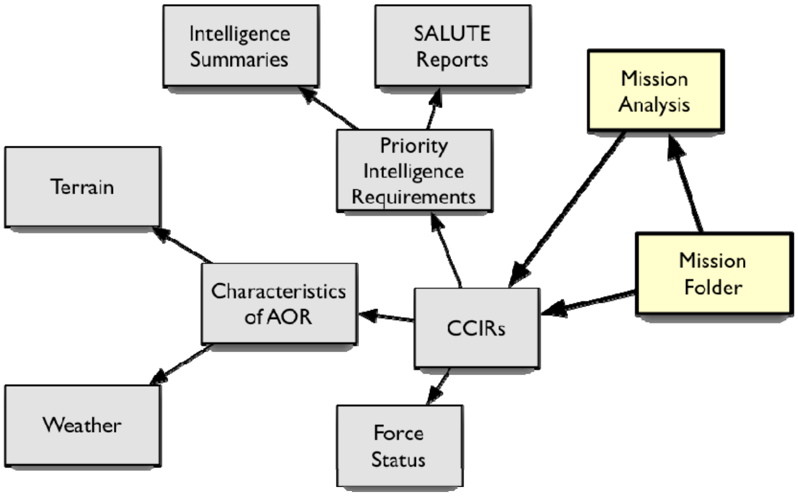


Figure 7: Step 2 - Definition of Mission Folder to Connect Mission Analysis with CCIRs

Figure 8 shows the third step, development of the Commander's Estimate and its addition to the mission folder. The Commander's Estimate provides updated mission analyses and intelligence estimates, own courses of action, as well as information, surveillance, and reconnaissance (ISR) plans. Note that as these new elements are added to the mission folder, existing elements may change based on new incoming intelligence reports, force status updates, weather conditions, etc.

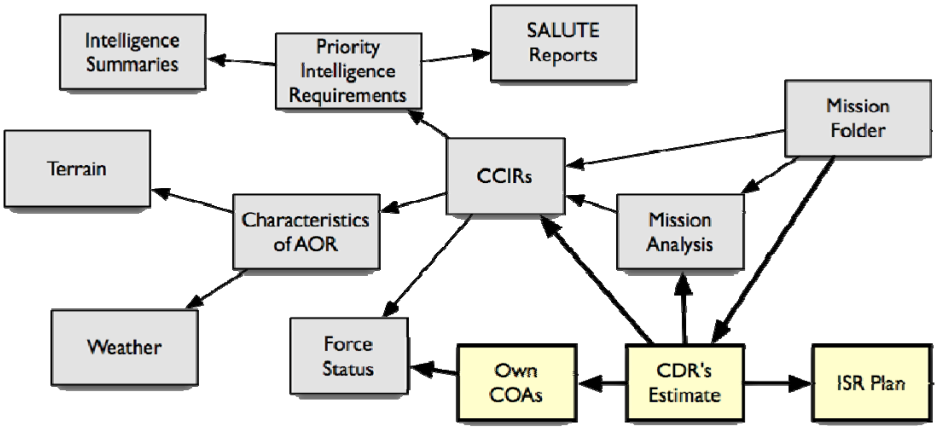


Figure 8: Step 3 - Development of Commander's Estimate and Proposed Courses of Action

After leadership selects a COA option for development into a full plan, detailed planning processes occur. This may entail development of primary plans and secondary plans, based on guidance established in the Commander's Estimate. Figure 9 illustrates how these products could integrate into the overall graph. As before, other elements in the graph could change as distributed participants inject updated intelligence reports, force status details, weather predictions, etc. However, what we see developing is a scale-free model of the operation, organized and structured by navigational nodes such as the mission folder, commander's estimate, and the CCIRs.

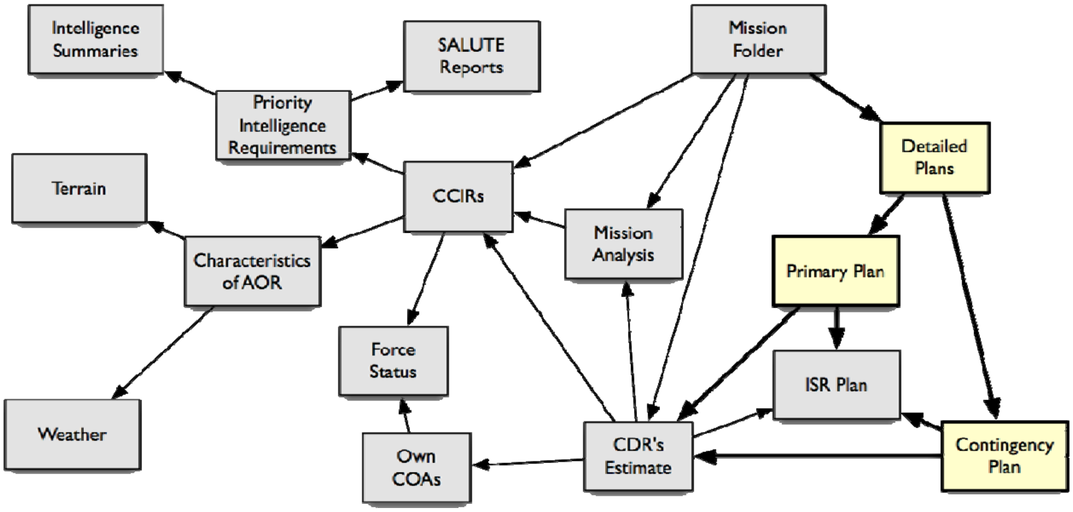


Figure 9: Step 4 - Development of Detailed Plans

The final step is mission execution, in which assigned forces carry out the planned mission in accordance with command intent. Key information products in this phase are monitoring and assessment of the ongoing operation in relation to those plans, to ascertain success or failure, and to identify any conditions that may call for modification of the original plans. Figure 10 presents a simplified view of this final step, in which execution monitoring information is attached to the mission folder.

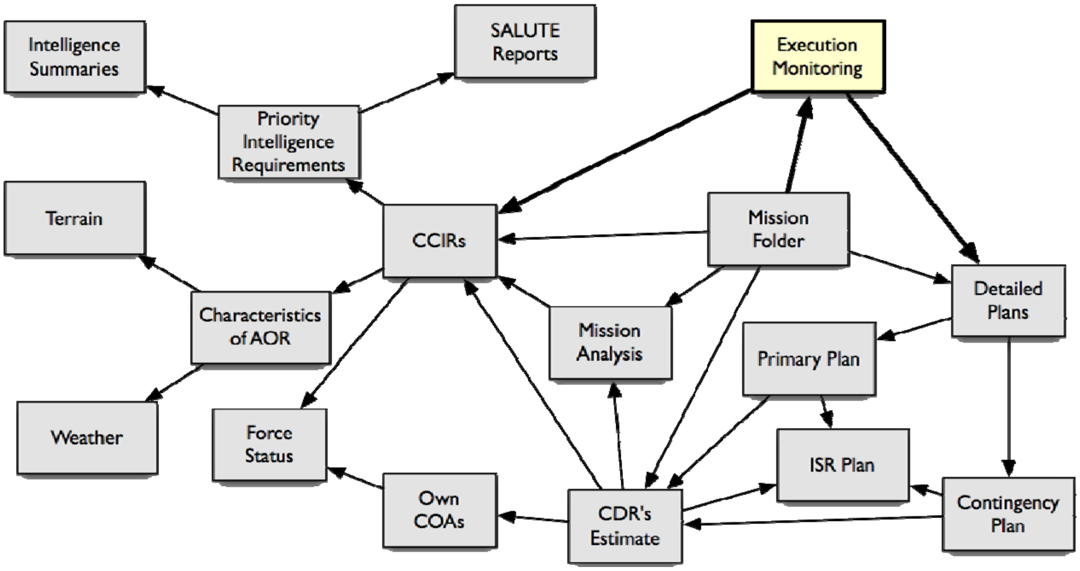


Figure 10: Step 5 - Integration of Execution Assessment Relative to Plans

Execution monitoring may contain an entire sub-graph of its own, which we omit from the picture for brevity. The final illustration shows a simplified view of the complete ensemble of information that would support a crisis action planning process, growing over time as the mission evolves and moves from one distinct phase to the next. Individual participants can augment the model to contribute individual elements of information (intelligence reports, force status updates, etc.), and they can also retrieve the information they need to support their own needs.

The example scenario described here illustrates how the proposed approach for modeling information requirements for agile C2 can work in practice. The graph structure lends itself to both predefined structure and organic growth. Different hub nodes within the graph act as focal points for different areas of information relevant to the command concept, and the links between them define critical relationships. We believe that it provides a scalable, flexible approach for supporting information management requirements that will be an essential component of the evolution to agile C2.

6. Summary

Military forces are being called upon to execute a broad spectrum of missions, with a desire to assemble, deploy, and make operational capability within mere days. Such requirements place enormous demands on the commanders executing C2, C2 processes, and C2 systems. A critical challenge in such agile capabilities is to ensure that all participants understand their roles, responsibilities, and relationships to peers, subordinates, and commanding C2 participants throughout an evolving mission. From an information management perspective, supporting C2 systems themselves must be agile and capable of accommodating change. A C2 system cannot be retooled every time new information needs are identified or as mission partners change. Building agility into a C2 system requires flexibility and foresight in how it creates, exposes, and updates the *business objects* that represent the essential information for C2. We have argued that systems designed to enable agile C2 must support three essential requirements in how they encode and make available these business objects:

- It must support changing the content and structure of the business object, as events evolve and new sources of information become important, new areas of focus develop, and dynamic relationships form between participating entities.
- It must be transparent to change, so that systems and users who are relying on it can detect and understand the impacts of those changes.
- It must afford different perspectives and views on the underlying data, so that each participant in a collaborative endeavor can access and use what they need for their mission, without having to be overburdened with unrelated elements.

Traditional systems attempt to be anticipatory of information needs and provide fixed data model designs that can support all envisioned use cases. Such systems attempt to expose all possible information to consumers, push all possible changes to underlying data, and give a finite set or single perspective of the information needed by its users. However, these systems will never anticipate all possible scenarios of use. As they are retooled to handle new situations not anticipated at design time, will become increasingly complex and brittle.

Instead, we propose an approach that values run-time flexibility over design-time anticipation, focusing on the essential information that C2 systems must model and share. We have presented a strategy for modeling C2 business objects that align with the principle of the command concept, which defines a vision of a prospective operation and all related supporting information. To encode this business object we borrow from the realm of network theory, which has shown that graph-oriented scale free networks provide a powerful paradigm for describing relationships in complex webs of knowledge. This graph modeling approach directly supports the requirements identified above: scale-free graphs can grow and change as needed, changes can be detected readily through changes to specific nodes and linkages, and different hubs within the graph can be used as anchors for different functional perspectives into the underlying data. Such an agile data modeling approach provides a strong foundation for evolving towards agile C2 capabilities and agile execution.

7. References

Alberts, D. & Hayes, R. (2006). *Understanding Command and Control*. Washington, D.C.: Command and Control Research Program.

Alberts, D. & Hayes, R. (2007). *Planning: Complex Endeavors*. Washington, D.C.: Command and Control Research Program.

Barabasi, A. (2003), *Linked.* Cambridge, MA: Plume.

Barabasi, A. (2007). “The Architecture of Complexity,” *IEEE Control Systems Magazine* (August).

Builder, Char H., Steven C. Banks and Richard Nording (1999). *Command Concepts*, Santa Monica CA: RAND Corporation.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001), "Web Service Description Language (WSDL) 1.1," World Wide Web Consortium (W3C). Available online at http://www.w3.org/TR/wsdl.

Davies, J., Schorow, D., Ray, S., and Rieber, D. (2008), *The Definitive Guide to SOA: Oracle Service Bus*, *Second Edition*. Apress.

Department of Defense (2003). "Department of Defense Net-Centric Data Strategy," Prepared by Department of Defense Chief Information Officer.

Endsley, M., Bolté, B., & Jones, D. (2003). *Designing for Situation Awareness*. New York: Taylor & Francis Group.

Joint Staff (2000). "The Joint Staff Officer's Guide 2000," National Defense University Joint Forces Staff College. Available online at http://www.jfsc.ndu.edu/current_students/documents_policies/documents/jsogpub_1_2000.pdf

Miller, W. (2004), "Cursor on Target", *Military Information Technology Online*, Volume 8, Issue 7 (Sept. 2).

Moffat, J. (2008). "The Response to Hurricane Katrina: A Case Study of Changing C2 Maturity," *Presented at the 13th International Command and Control Research and Technology Symposium*, Seattle, WA.

Mulgund, S. and Landsman, S. (2007). "User Defined Operational Pictures for Tailored Situation Awareness," *Presented at the 12th International Command and Control Research and Technology Symposium*, Newport RI.