Cover Sheet

Engineering Model For Enterprise Command and Control

Jay Bayne †
Echelon 4 Corp
414.617.5595
jbayne@echelon4.com
† Primary Contact

Raymond Paul
OASD/NII
703.607.0649
raymond.paul@osd.mil

15 March 2004

Engineering Model For
Enterprise Command and Control in Federated Systems

Jay Bayne †
Echelon 4 Corp
414.617.5595
jbayne@echelon4.com
† Primary Contact

Raymond Paul
OASD/NII
703.607.0649
raymond.paul@osd.mil

Abstract

Enterprises are increasingly complex; especially those found in National Defense. They are dynamic systems, geographically and operationally distributed, and faced with operating with increased agility in "always on" conditions under greater regulatory scrutiny within highly volatile "market" conditions where continuous quality improvements are expected. This situation requires continuous improvements in the conduct of enterprise operations, with a corresponding increase in the efficacy of decision (command) and control applied to all facets of enterprise management. To address this challenge we introduce and engineering model supporting advances in both the concept and implementation of *enterprise command and control* (EC2) systems. We characterize the objectives of EC2 in terms of *units of value production* (VPU) that function at the intersection of enterprise *supply* and *asset chains*. Enterprise governance is consequently focused on the continuous optimization of performance of value production processes linked by these two value chains. VPUs range in size and function from low level tactical, to mid-level operational, to high level strategic activities, all of which must interact effectively in order to sustain enterprise viability. This paper provides a summary of our work in the application of our VPU model to the EC2 requirements of complex federated enterprise systems.

Key Words: Enterprise, Command and Control, Systems Engineering, Enterprise Systems, Cybernetics, Agile Systems, Reactive Systems, Real-time Systems

1 Enterprise Value Production

Our thesis is that effective (e.g., agile and error free) governance of large-scale distribute, always on, real-time enterprise[1], in response to probabilistic environmental and self-imposed demands, requires the *systematic* (*engineered*) application of an intelligent decision and control system framework[2]. To improve the efficacy of management we are presently designing advanced software systems to assist in real-time distributed enterprise decision (command) and control (EC2). Doing so requires a concise operational definition of enterprise and an identification of its core *value production processes,* processes that can benefit from improvements in automation and control.

---

[1] The term "enterprise" refers to an arbitrary but named *unit of organization* whose existence is expressly for the stable production of a quantifiable measure of *value*.
[2] The term "framework" refers to an *architectural pattern*, the definition of a complete, coherent and self-consistent *template* for creation of applications within a given domain.

To maintain its viability[3] an enterprise must be *dynamically stable*[4]. Stability requires that it offer a sustainable and competitive *value proposition* to the evolving environment (e.g., "marketplace" or "battlespace") within which it operates. A value proposition defines the ratio between the costs to provide a benefit to a domain and the domain's continuous revaluation (e.g., "clearing price") for that benefit. This requires enterprise to be agile (adaptive) with respect to changing domain conditions.

The difference between fully burdened value production costs and domain clearing prices, measured in the domain's economic units, equals the marginal benefit (*profit*) realized by the enterprise in the continuous evaluation (*execution*) its value propositions. An enterprise is *viable* to the extent this marginal benefit is both sustainable and sufficient to fuel adaptation within its competitive environment. In other words,

> *A viable enterprise is a computational object (virtual machine) that continuously executes a finite set of adaptive programs (its value propositions) whose results provide marginal benefits sufficient to 1) satisfy its evolving market requirements and 2) fuel internal innovations sufficient to maintain homeostasis.*

Enterprise decision and control is about responding to evolving operating *situations* through development of appropriate *responses*. Responses define the continuous improvements to core value production processes required for maintaining viability. To quantify and continuously improve value production, we need a model of a *value production unit* (VPU).

1.1 Units of Value Production

The locus of enterprise management activity responsible for production of a quantifiable measure of value is referred to as a *value production unit* (VPU). A VPU is an abstract object[5] that participates in *production webs* with other VPUs. As drawn in Figure 1 webs are bound by *value chains*, specifically a vertical *asset chain* and a horizontal *supply chain*.

A VPU is uniquely identified by a relative address within the web. In Figure 1 VPU[k,l] identifies a value production process at the "lth" level in an asset chain and the "kth" position in a supply chain. VPU[k,l] is *subordinate*, and therefore accountable to, VPU[k,l+1] in the asset chain, and a *server* or service provider, and therefore committed to, VPU[k+1, l] in the supply chain. Likewise, VPU[k,l] is a *superior* to, and therefore responsible for, VPU[k,l-1] in the asset chain, and a *client* of, and therefore dependent on, VPU[k-1,l] in the supply chain.

A VPU may simultaneously participate in a number of *non-conflicting* webs.[6] VPUs interconnect through four sets of duplex communications ports, two each for the two value

---

[3] A system is "viable" to the extent it maintains its existence over time.
[4] In natural systems, dynamic stability in the face of evolutionary pressures is referred to as *homeostasis*.
[5] A VPU is an enterprise *virtual machine* representing (i.e., a *proxy* for) one or more real enterprise activities that results in a specific and quantifiable degree of value production.
[6] For this analysis we will consider only single vertical and horizontal VPU dependencies. Issues of VPU fan-out and fan-in are treated elsewhere.

chains. Arrows associated with each port indicate the direction of the *flow of increasing value*. The function of each port is summarized in Table 1.

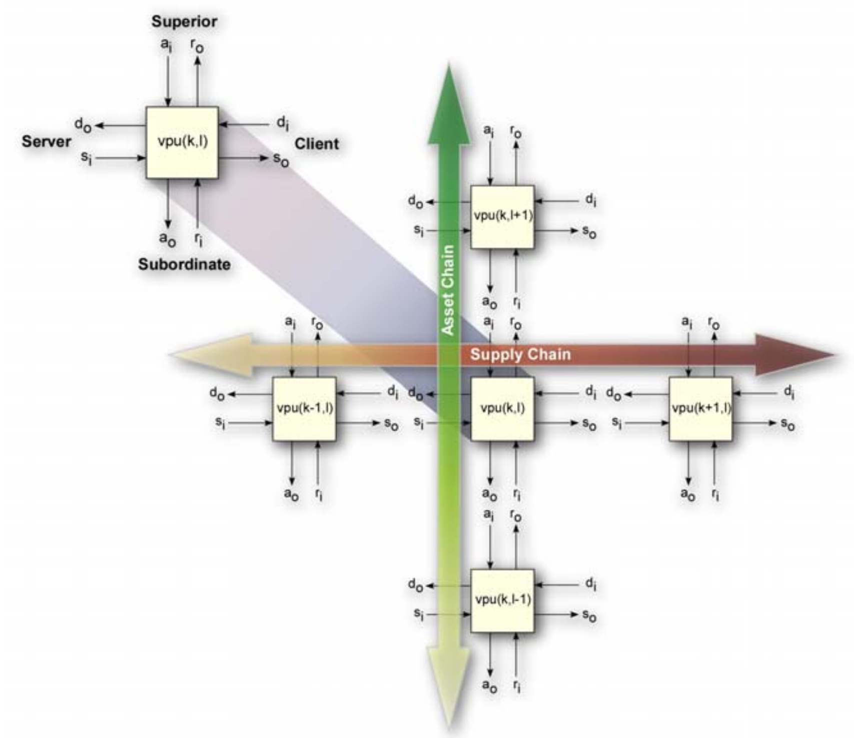

Figure 1 - Value Product Units (VPU)

Table 1 – VPU Communications

Value Chain	Port ID	Port Name	Port Function
Asset Chain	a_i	Assets In	Acceptance and assimilation, according to a service-level agreement (SLA), of allocated assets from superior VPUs
	r_o	Returns Out	Production of returns on value produced by previously allocated assets; requests for allocation of additional assets
	a_o	Assets Out	Allocation, based on a SLA, of assets to subordinate VPUs with expectations for a minimum time-specific return of value for the allocation
	r_i	Returns In	Acceptance and assimilation of returns and evaluation of requests for asset allocations from subordinate VPUs
Supply Chain	d_i	Demand In	Acceptance of demands (orders) for goods or services from upstream consumer (client) VPUs
	s_o	Supply Out	Fulfillment of demand (orders) in the form of goods or services to downstream consumer (client) VPUs
	d_o	Demand Out	Issuance of demands (orders) for goods or services to upstream producer (server) VPUs
	s_i	Supply In	Acceptance of fulfilled orders for goods or services from downstream producer (server) VPUs

This characterization of value production identifies the core computational requirement of a VPU, and therefore the principle objective of decision and control, as *the simultaneous and continuous (i.e., real-time) optimization of value production on both intersecting value chains*. In the web framework this is both a local (internal, closed, private) optimization problem performed within a VPU, and a global (external, open, public) optimization among neighbors within the lattice. Governance (EC2) of an enterprise may therefore be viewed as the individual and federated management of populations of interacting VPUs.

2 Enterprise Command and Control (EC2) Model

In adaptive systems, coordinated local and global optimization through feedback control mechanisms is a generally complex activity, even for relatively simple processes operating in static environments. Enterprises and their embedded value production processes are far from simple, either individually or in ensemble. Furthermore, they operate in dynamic environments where adaptation is a key determinant of survival through sustainable value production.

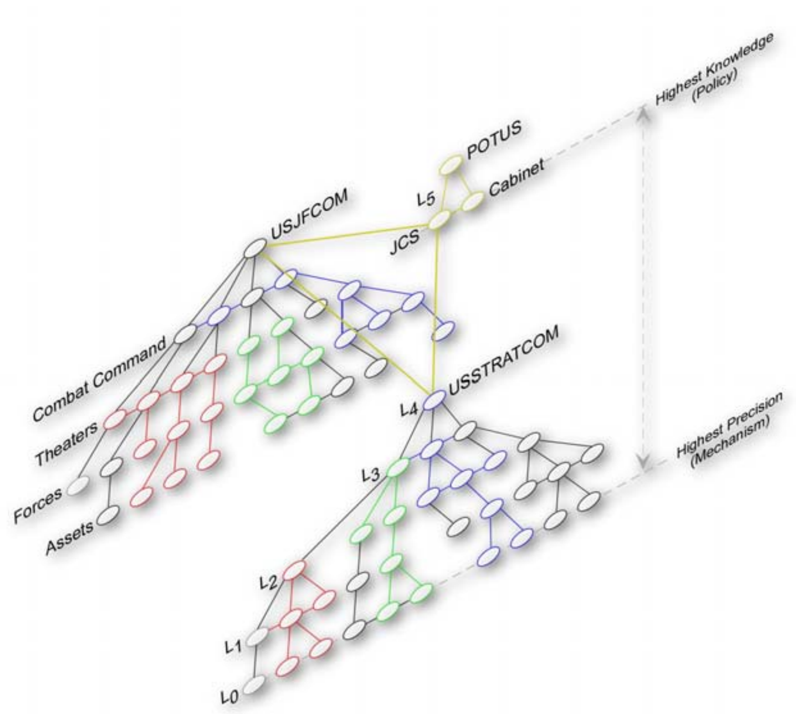

Figure 2 - Enterprise Policy Domains (Trees)[7]

Enterprise C2 must therefore serve a range of value production management requirements, from control of short-term, highly dynamic, internal *tactical* activities to longer-term, more slowly evolving, global *strategic* activities. In the middle of this spectrum are the more methodical, mundane and pragmatic *operational* activities that represent the core of enterprise behavior.

[7] The figure represents the scope of the US DOD Unified Command Structure (UCS), but could equally we represent the structure of medium to large scale commercial enterprise.

Identifying these activities and striking a balance between tactical, operational and strategic initiatives is critical to survival.

We begin to quantify these behaviors by describing the nature of asset chains – behaviors in the well-known and often maligned *policy* or *accountability hierarchy* of enterprise. Figure 2 diagrams a six level enterprise structure[8]. The figure represents the US National Command Structure, including the President of the US (POTUS), the Cabinet and Joint Chiefs of Staff at Level 5, the US Joint Forces and Strategic Commands at Level 4, down to men, machines and material at Level 0. To represent commercial enterprise, the levels in this figure could equally well be labeled with business terminology (business areas, units, production plants, etc), representing management domains where generally accepted accounting practices (GAAP) are applied, where fiduciary accountability must be maintained to sustain equity market viability, and where causal relations must be maintained in order to sustain regulatory and legal viability.

2.1 EC2 Policy Framework

The top level node in Figure 2 defines the root of a *policy domain tree*, where each subordinate node represents, in a recursive fashion, the root of a subordinate or embedded policy domain. Policy domains define regions where enterprise decision and control action is governed (constrained) by policies or doctrines that relate to domain-specific *value propositions*[9]. Policies express ethical, political, legal, financial, temporal or other conditions under which VPUs must operate, individually and in ensemble.

Each node in a policy domain tree represents one or more VPUs, with their vertical links defining an asset chain and their horizontal links defining supply chains. Figure 3 diagrams value chains as might exist within and between two hypothetical allied military enterprises, identified here as “A” and “B.” They have defined a high-level *inter-domain* supply chain agreement (“coalition”) denoted by the dashed line linking their respective Level 5 policy domains. Within military A there is an internal Level 3 supply chain between its B2 and B3 policy domains. There are intra-coalition Level 1 and Level 3 supply chains between the A1 and B3.

The normal and routine functioning of enterprise consists of decision and control occurring throughout its policy domain hierarchy, often with only loose social concepts of EC2. This essentially ad hoc approach to management is a principle source of tactical, operational and strategic errors and inefficiency, resulting in slow response (loss of agility) and lost productivity.

[8] Six is somewhat arbitrary number, but serves to emphasize a (typical) management structure of medium to large scale enterprise.

[9] A *value proposition* is a predicate specification of the general form:

```
if <situation> and <response> then <expected_result>
```

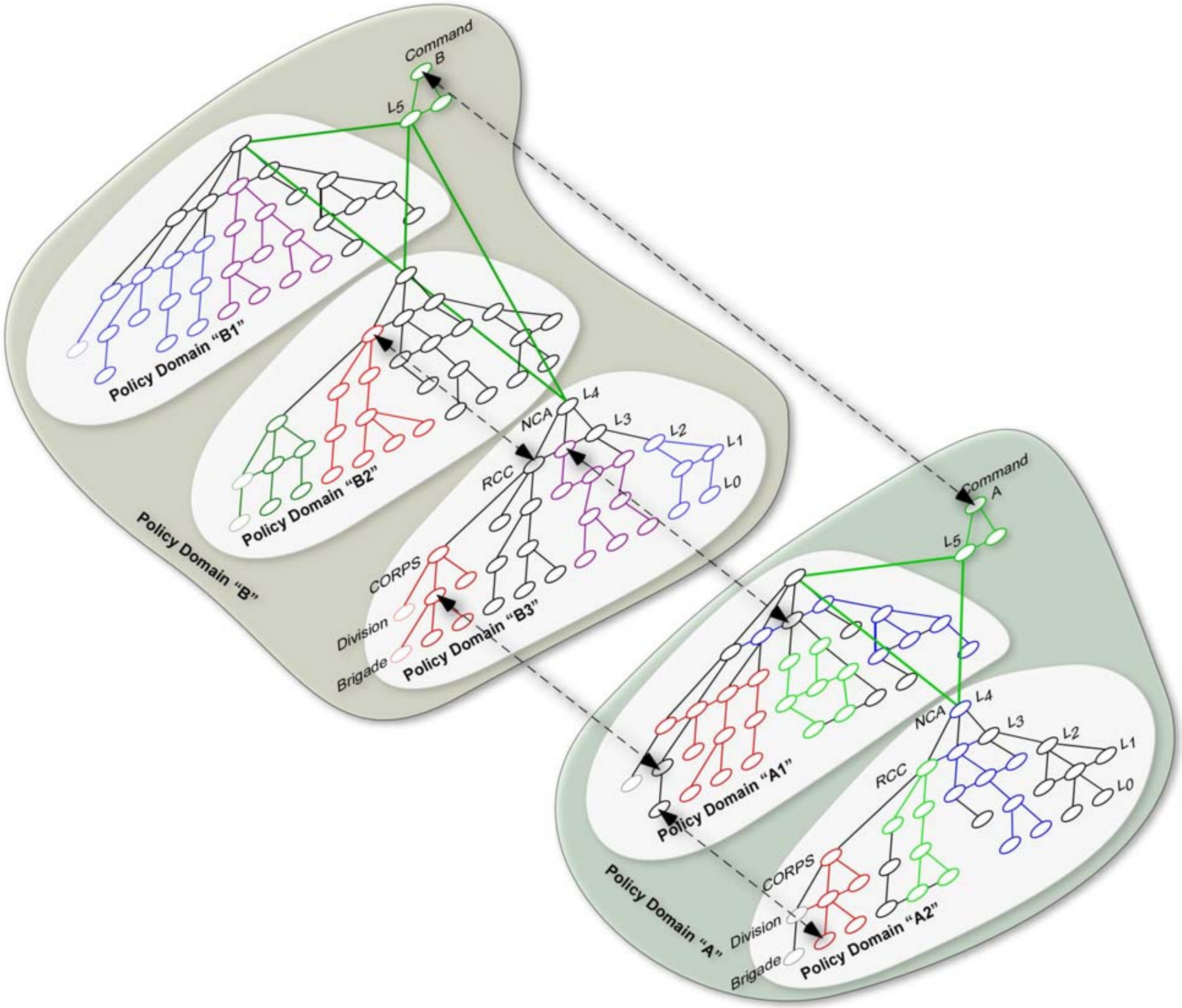

Figure 3 – Inter-Enterprise Supply Chains

At its simplest,

> *Enterprise command and control is the exercise of authority and direction. In general, EC2 is the real-time exercise of policy-constrained authority and direction, guided by a manager's intent (command) and accomplished through an adaptable, decentralized, and cross-organizational arrangement of capabilities (i.e., personnel, services, communications and equipment) that are inter-connected and that operate through a common shared information infrastructure (control).*

While intuitive and useful, this definition is far from prescriptive on the act of "doing C2." A more operational definition would necessarily include description of the activity or *process of*

C2. Figure 4 presents a classical[10] C2 process with its generalized services. The figure identifies, beginning at the bottom, a value production process under control (PUC). The process control loop begins with one or more measurements of the status (i.e., actual or inferred behavior) of the process[11]. These measurements collectively feed the activity of sensory perception, qualified by policy-based value judgments and historical trend information stored in a process model (i.e., knowledge base). Together these activities constitute the continuous act of maintaining a degree of *situational awareness*. Once appropriately aware of the current situation the activities of behavior generation (planning and model update) and execution (resource management and final control) complete the control loop. This cycle may operate continuously, be periodically invoked, or be aperiodic, running asynchronously only when measurements generate perceptions that exceed some specific threshold and trigger a C2 reaction. The inner loop of the C2 process, when automated, provides a degree of self-regulatory or *autonomic control*. The outer loop, generally in the heads and hands of human beings, provides various degrees of *adaptive supervisory control*.

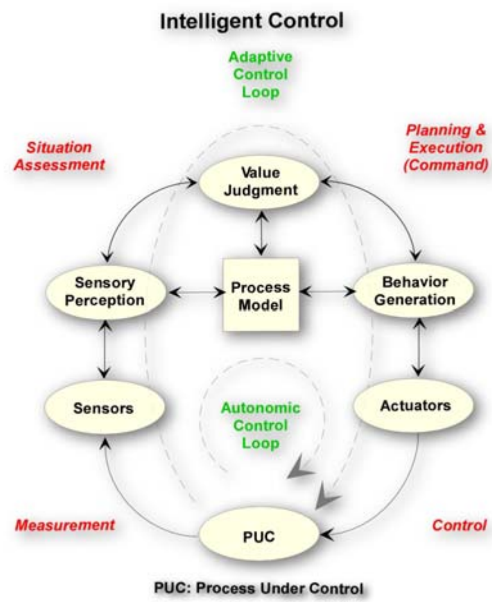

Figure 4 – The Process of EC2

The model is nested, with each node in an accountability hierarchy executing this C2 process (in an appropriate form) and treating subordinate VPU nodes as their respective "processes under control." Similarly, each policy domain is *embedded* in a higher-level domain. Figure 5 expresses the important point that EC2 is a process that happens throughout the policy domain hierarchy, concurrently, within each VPU. Control derives not from a centralized function, but rather from the collective behavior of federations of collaborative C2 processes – distributed control, operating under a cascading set of policies. Coherence of behavior is achieved through 1) nested policy domains and 2) collaboration along value chains among cooperating VPUs.

[10] The figure is a classic model of feedback control in adaptive systems, and at the heart of the science of cybernetics – automation and control in complex dynamic systems.
[11] Measurements may be made synchronously (sampled) or asynchronously (event driven).

Ideally, in supporting scale economics and the desire for validated (quality) and reusable process steps, the EC2 process is implemented at each node using common underlying methods. Figure 5 expresses this ideal situation by implying that tactical C2 at the base of the policy tree and strategic C2 at the top are both carried out, simultaneously, using the same C2 process model. Each node in a policy tree is a virtual machine that executes the C2 process on behalf of its value propositions.

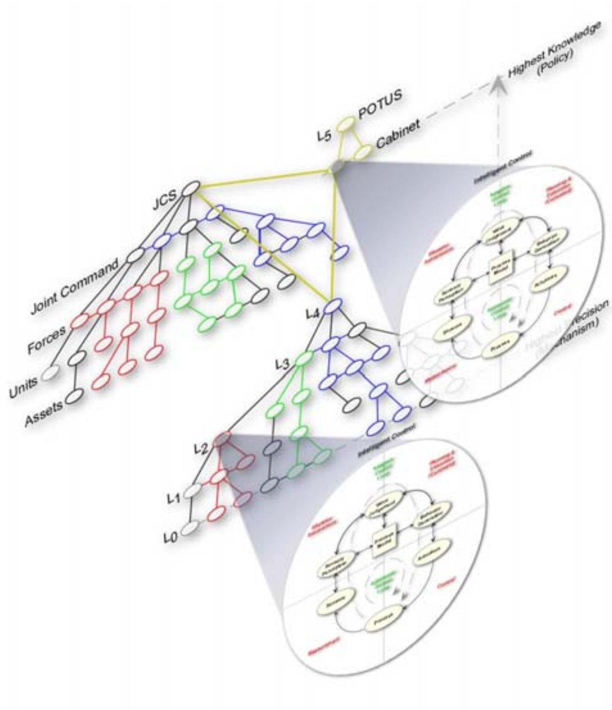

Figure 5 – Pervasive, Adaptive and "Always-On" EC2

Figure 2, Figure 4 and Figure 5 taken together express two fundamental challenges of EC2 in distributed systems: simultaneously managing *intra-domain* and *inter-domain* value production among cooperating VPUs (i.e., *communities of interest*, COI). This is the all too familiar societal, commercial and sovereign governmental dilemma of managing the competing interests of the private (internal, personal) and public (external, allied) aspects of value production. The collaborative management of public (i.e., federated) domain value production requires a *policy commons* among cooperating entities. Such a commons provides the basis of service level agreements (SLA) that govern value chain *contracts* among the federated VPUs.

2.2 Unified Enterprise C2 Structure

Within a policy domain hierarchy the process of adaptive C2 must be governed in order to manage inheritance of value propositions and their associated policy specifications. As noted, managing value production within and across enterprise policy domains requires existence of 1) one or more value production *processes*, 2) well defined and scalable machinery for implementing consistent C2 measurement and control processes (Figure 4), and 3) formal *user interfaces* for engaging responsible human management *actors* engaged in both private and public value production. Taken together, these three elements comprise our Enterprise Control Model (ECM), shown in Figure 6.

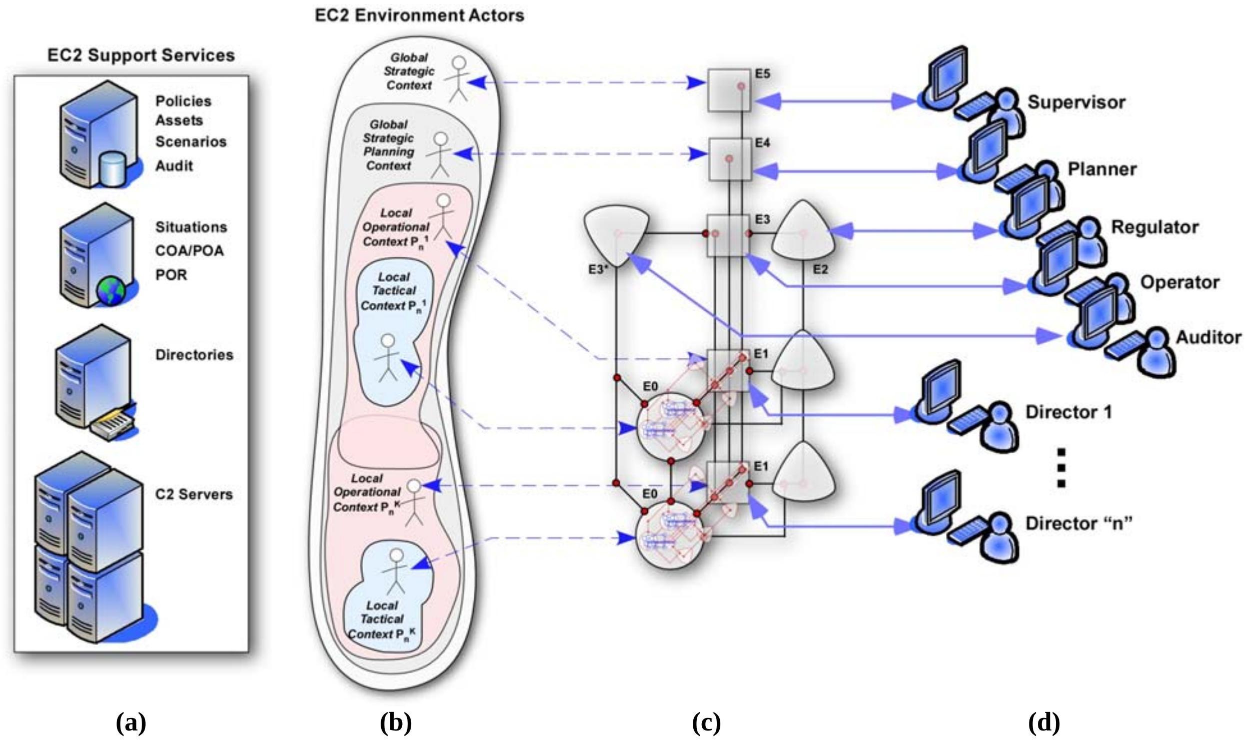

Figure 6 - EC2 Actor Model

At the core of the C2 challenge is provision of *coordinated* automation and control. Enterprise management is responsible for materially improving the efficacy of value production, especially in supporting efficient and reliable collaboration among (i.e., distributed and federated) human enterprise actors. Simply stated, EC2 is a management *service* within an enterprise. Processes deliver services (i.e., execute service applications). The *structure* of C2 processes (*actors*) is diagrammed in Figure 6c, the components of which are outlined in Table 2. Each component represents an *echelon*[12] (E) of control. Governance by *actors* requires an *organization* in order to coordinate the behavior of the actors and to manage the acquisition and disposition of resources each requires.

Within a given policy domain it is typical to find several governance roles, each defined by a generalized set of responsibilities. Again referring to Figure 6c and Table 2, the organizational activities and their interrelationships include: at echelon 5 (E5), the highest level of authority, domain management (supervision); at echelon 4 (E4), the analysis, planning and development functions for managing adaptation; at echelon 3 (E3), the operations directorate for executing plans of action, including its audit function (E3*); at echelon 2 (E2), regulatory controls needed for coordinating parallel task that cooperate (rendezvous) and access shared resources; at echelon 1 (E1), the task directors that manage specific value production processes and their interaction

[12] The term "echelon" refers to the level, rank or authority of an entity.

with value chains (P_n) at E0. Here "n," the number of such subordinate value production processes, is equal to or greater than one, and typically less than 10.

Table 2 – Principle Enterprise C2 Processes

Echelon	Service Name	Enterprise Roles & Responsibilities
E5	Supervision	Goals, Objectives & Policy Domain Management
E4	Planning	Mission Capability Management
E3	Operations	Program & Capability Management
E3*	Auditing	Program and Process Performance Assessment
E2	Control	Process (Task) Synchronization
E1	Command	Process (Task) Management
E0	Process	Value Production Process (Task)

Note: "*" designates a subordinate role at a given echelon.

Note in Figure 6c that the management organization for E0 VPUs is structured, at the next lower level of the policy tree, identically to it's parent (supervising, containing) VPU. This nested (recursive) model is a key element in our distributed ESM model, and is consistent with the goal of designing service-oriented EC2 solutions that scale.

The recursive structure of the ECM controller model is diagrammed in Figure 7. Notice that as we look deeper into the organization, at each successive level in the policy domain we find the same management structure. This facilitates designing and deploying a common set of C2 services that can be used throughout the enterprise. Without such a symmetric and recursive control model, each domain would have to create its own machinery for C2 and force linkages to other domains in a federation.

2.3 Enterprise C2 Actors

Within complex enterprise systems, each of the process control entities mentioned is *managed* (administered, supervised, directed, governed, owned) by a human *actor*. These actors are diagrammed in Figure 6d, and summarized in Table 3, and are key elements of C2 services since they *drive* the C2 applications. They represent the captain (pilot) and crew of the enterprise VPU.

The process model shown is derived from cybernetics[13], and consists of familiar management functions found in naturally occurring and man-made systems. For any given activity there is a superior authority responsible for its conduct (E5); there is an actor responsible for sensing impending changes in the environment containing the activity and planning for and developing consequential adaptations required to maintain enterprise efficacy (E4); there is an actor responsible for executing planned and resourced actions (E3); there is an actor responsible for assessing the performance of currently executing activities (E3*); there are actors responsible for carrying out tasks specified in the E3 plans (E1); and there are actors responsible for coordinating parallel and interdependent tasks (E2) when there are multiple E1/E0 tasks. The behavior of each echelon controller (i.e., the VPU object's *methods*), and the protocols used to link them, are the subject of an Echelon 4 Inc. design document.

[13] The term "cybernetics" refers to the science of dynamic systems, and the technical space where computing, communications and automatic (feedback) controls intersect.

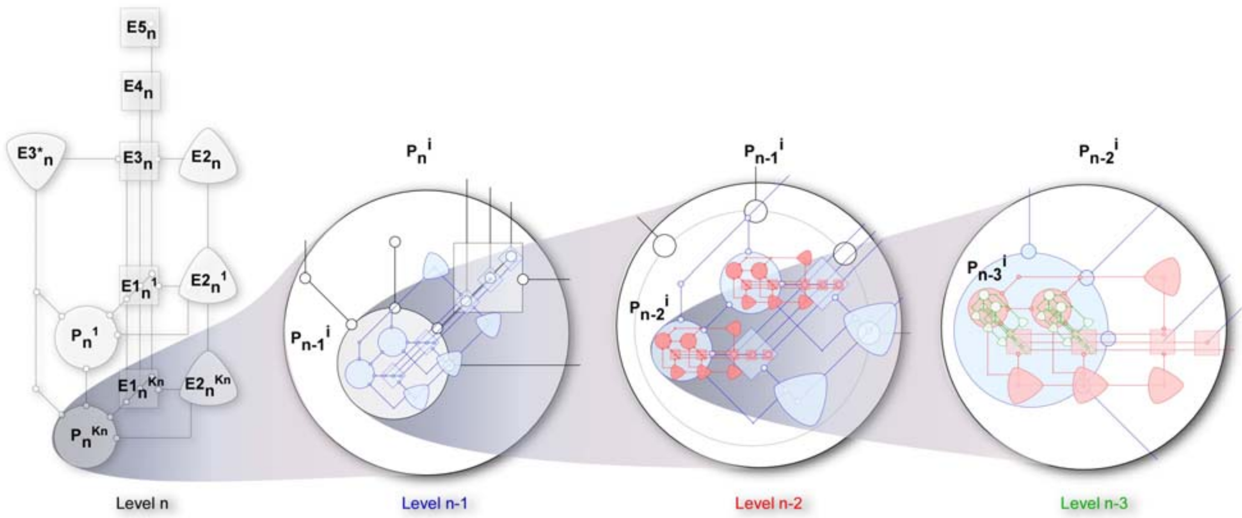

Figure 7 – Recursive EC2 Actor Model

Table 3 – Principle EC2 Actors

Actor	Actor Name	Count	Enterprise Actor Roles
E5	Supervisor	1	Policy Domain Manager, Policy Management
E4	Planner	2	Mission Capabilities Manager, Enterprise Systems Engineering
E3	Operator	1	Program Manager, Operations
E3*	Auditor	1	Program and Process Performance Assessment Manager
E2	Controller	N≥1	Per-process (Task) Synchronization Manager
E1	Commander	N≥1	Per-process (Task) Manager
E0	Producer	N≥1	Value Production Task

Note: "*" designates a subordinate role at a given echelon.

2.3.1 Enterprise Operating Context

Enterprise is embedded in and dependent upon a specific operating environment or context, e.g., an economic marketplace, a military battlespace, a healthcare concern. This context provides input to and receives output from the value production processes of contained VPUs. As shown in Figure 6b, the context is further partitioned into specific, typically overlapping, regions where C2 processes are focused. There exist regions of the environment that support strategic planning and provide insight into possible future situations, identified with E4. There are regions, identified with E3, where the VPUs' aggregate behavior is focused and where its operating conditions are unfolding. And there are sub regions of the environment, identified with the E1-E0, where the individual subordinate VPUs operate in tactical ways to reach their targets. This containment hierarchy corresponds to the policy domains of Figure 2.

Figure 6a shows shared support services (aka, "information infrastructure") required for managing the conduct of integrated enterprise behavior. The figure identifies a few of the key grid-connected information systems. They include repositories for the enterprise's operating policies, reusable and consumable assets, and validated scenarios (aka, defined business processes). In addition, the infrastructure supports storage of past and present of situations encountered, past and present plans of action (POA – feasible responses to situations), and current in-process plans of record (POR – resourced and in-process commitments to current situations). These information assets provide the basis for coordinated action, discussed below.

2.3.2 Enterprise C2 Application

The decision and control *application program* executed by the enterprise echelon controllers is diagrammed in Figure 8, with its major components summarized in Table 4. The application's inputs are measures of the present state of its environment (*situations*) and whose outputs are the VPU's responses to these situations in the form of executable plans of record (POR).

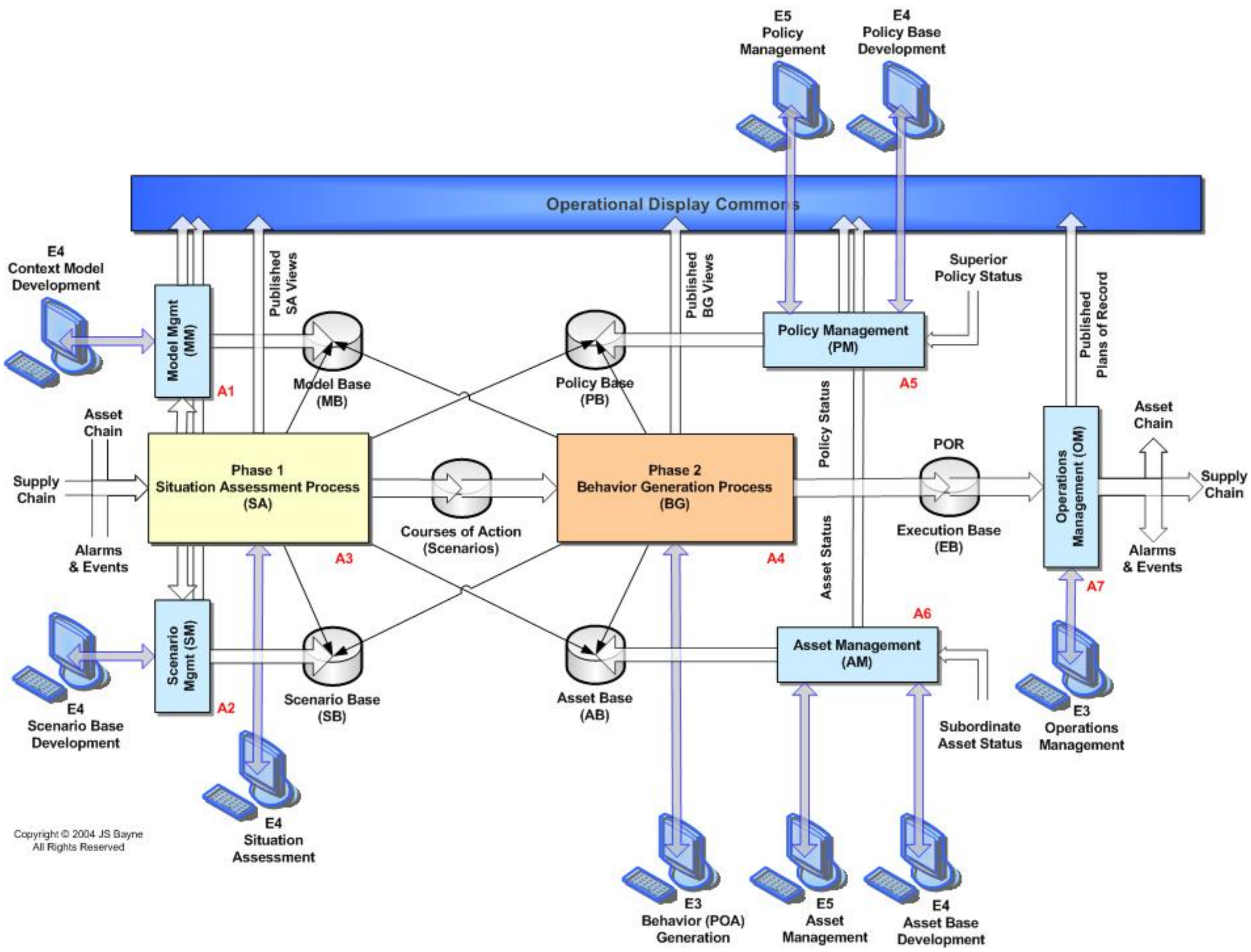

Figure 8 – EC2 Application Components

The program behaves as follows. Situations, in the form of periodic measurement subscriptions, asynchronous alarms and events, arising internally and in the value chains are consumed by A3, the situation assessment (SA) process. The process uses the context model database (MB), the validated scenario database (SB), the asset database (AB), and the policy database (PB) to triage and analyze current situations. The outputs of SA are scenarios representing potential courses of action (COA) that respond to current situations. The COA are subsequently input to A4, the behavior generation process (BG). BG uses MB, SB, AB, and PB to triage and prioritize COA against currently executing plans in order to convert COA into coordinated plans of action

(POA). The final step of BG is to assign, when available, required resources to POA converting them to executable plans of record (POR).

Figure 8 also identifies five supporting application components. A1 and A2 provide interactive management of the model and scenario databases, respectively. A5 and A6 provide interactive management of the policy and asset databases, respectively. And POR are put into execution and monitored by A7. As a final note, the principle actors introduced in Figure 6 are shown in relation to the seven application components.

Table 4 – EC2 Application Components (ref: Figure 8)

Component	Principle Actors	Function	Input	Output
A1 Model Management	E4	Enterprise Model Development	Subscription-based real-time operating situations and events; policy constraints; asset constraints; current situation model	Updated Model Database (MB)
A2 Scenario Management	E4	Enterprise Scenario Development	Policy base updates; asset base status updates; validated scenarios; current aggregate situation model	Updated Scenario Database (SD)
A3 Situation Assessment	E5, E4	Enterprise Situation Generation	Current domain capabilities; prioritized course of action (COI); current plans of record (POR)	Prioritized list of potential courses of action (COA) with resource requirements and policy issues; updated context situation model
A4 Behavior Generation	E3	Enterprise Behavior Generation	Prioritized Courses of Action (COA)	Updated set of resourced and prioritized plans of record (POR)
A5 Policy Management	E5	Enterprise Policy Management	Current policy status; current asset status; current COA status; current POR status; superior's policy status	Updated domain Policy Database (PB)
A6 Asset Management	E3, E4	Enterprise Asset Management	Current asset status; current policy status; current POR status; unmet COA requirements; subordinates' asset status	Updated domain Asset Database (AB)
A7 Operations Management	E3	Enterprise Operations Management	Pending POR	Sequenced execution of POR

2.3.3 Collaborating VPUs

VPU's must collaborate in two dimensions, simultaneously. They participate in their vertical asset chain and horizontally in their supply chain engagements. Policies and associated communications protocols differ with respect to value chain C2 requirements.

We noted in Section 2.1 that policy domains are defined by both peer-peer and superior-subordinate relations. The superior-subordinate dimension establishes the command accountability structure of enterprise. If our C2 solution is to scale effectively, is to be economical in implementation, and is capable of being validated for correct and dependable operation. It must not depend explicitly on the VPU's level in the policy tree or its position in the supply chain. This domain-neutral requirement limits options for constructing ad hoc policies and mechanisms for collaboration control. Our solution is based on the recursive nature of containment within accountability structures.

In Figure 9 three VPUs are engaged in (one of potentially many) collaborations related to their interdependent asset and supply chain activities. The Asset Chain Director in VPU[k,l] is in collaboration with her counterpart in subordinate VPU[k,l-1]. At the same time, the Supply Chain Director in VPU[k,l] is in collaboration with his peer-level supplier VPU[k-1,l]. Within VPU[k,l] it is the responsibility of the Operator to provide and coordinate plans (of record) for the Asset and Supply Chain Directors that meet the objectives of the Supervisor and her Planner.

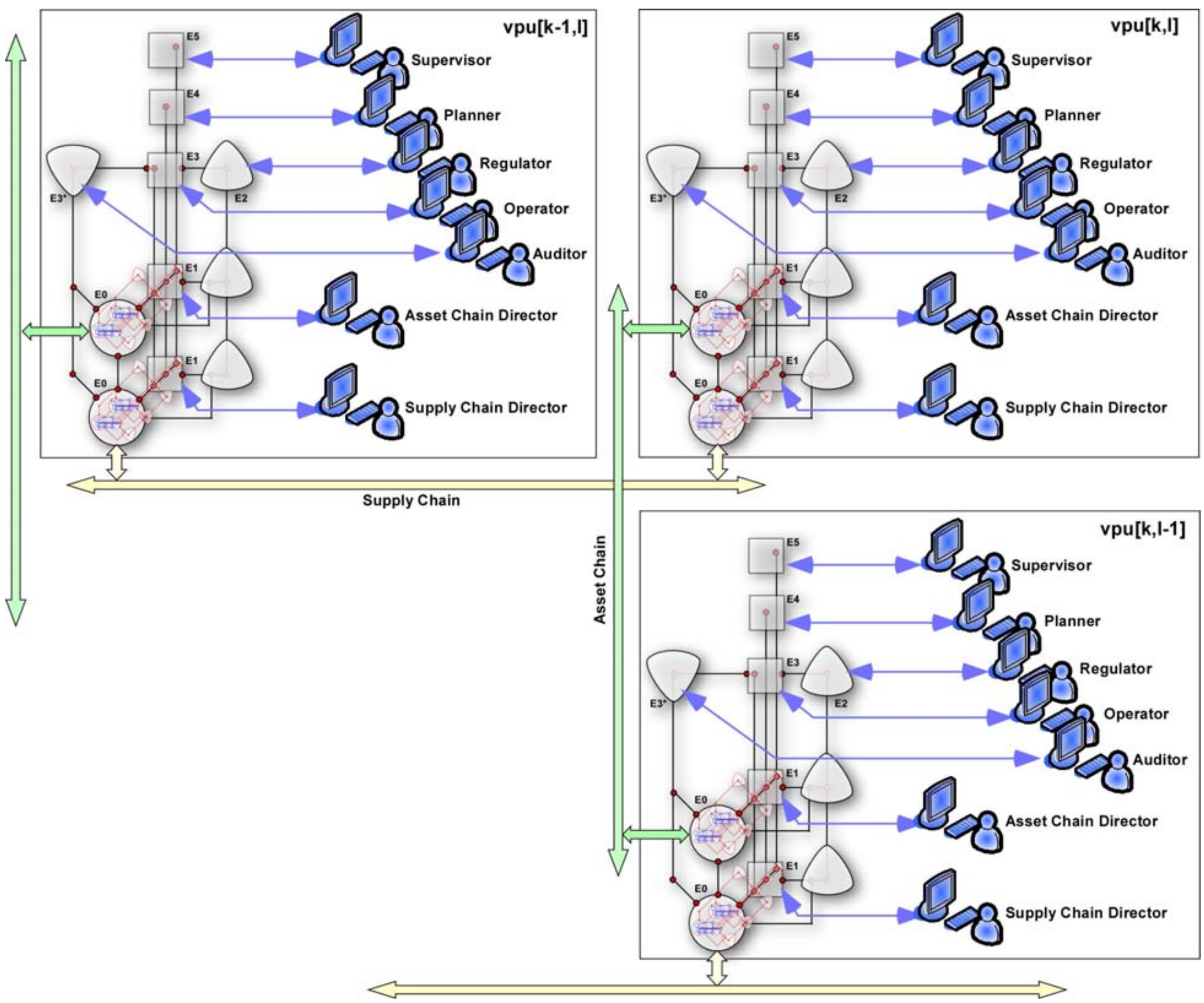

Figure 9 – Collaboration among EC2 Domains

The Asset and Supply Chain Directors are responsible for their E0 asset and supply chain value production processes, respectively. As discussed, each of these E0 processes may contain nested VPUs, as described in Figure 7. The nesting may, and likely will, contain lower level asset and supply chain relationships. This situation motivates the design and development of generalized (e.g., scalable, object- and grid-based) EC2 services that can be deployed at all levels of the enterprise.

To facilitate discussion of the collaboration between VPUs in Figure 9, we have drawn Figure 10, a diagram that offers a shared "whiteboard" metaphor on which situation assessment and plan execution can take place among the various actors. The whiteboard represents the multi-media environment on which the shared activities of Figure 8 are played out between cooperating VPUs (i.e., members of a given community of interest).

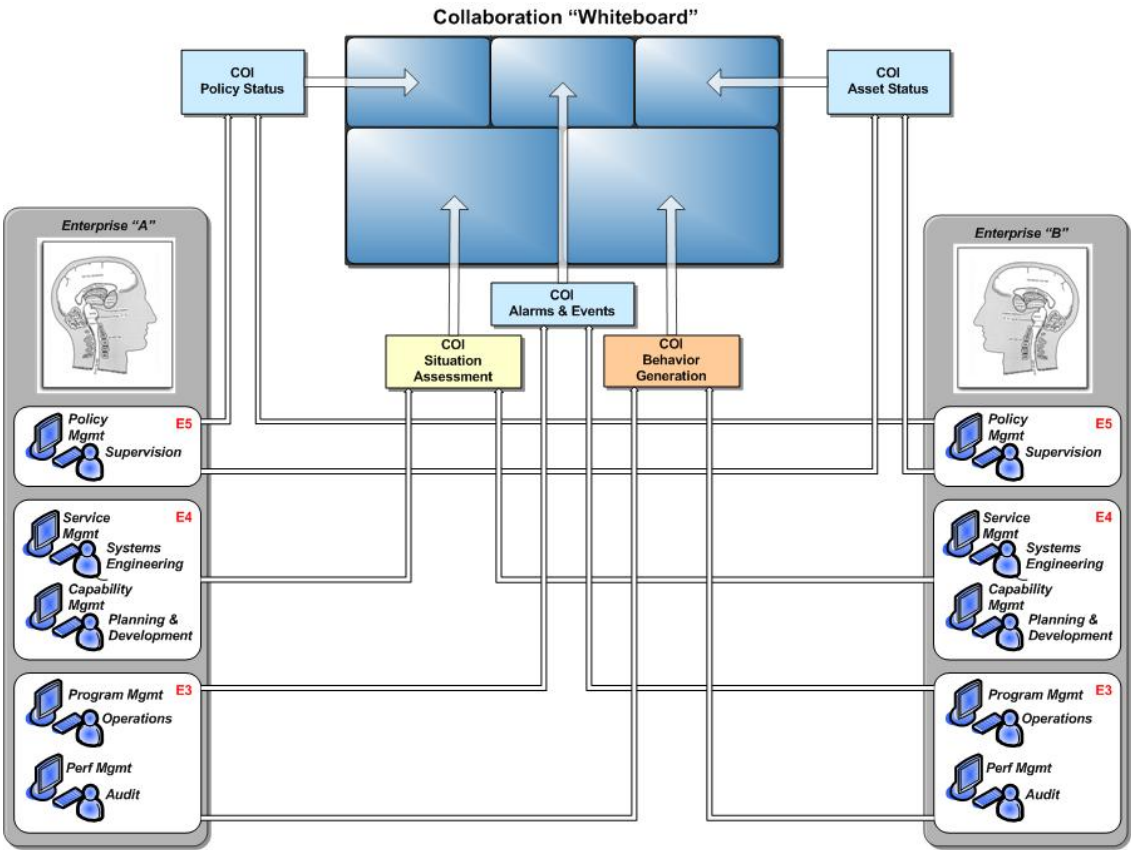

Figure 10 – EC2 Communities of Interest

Figure 11 - DOD USSTRATCOM Command Center Concept Drawings

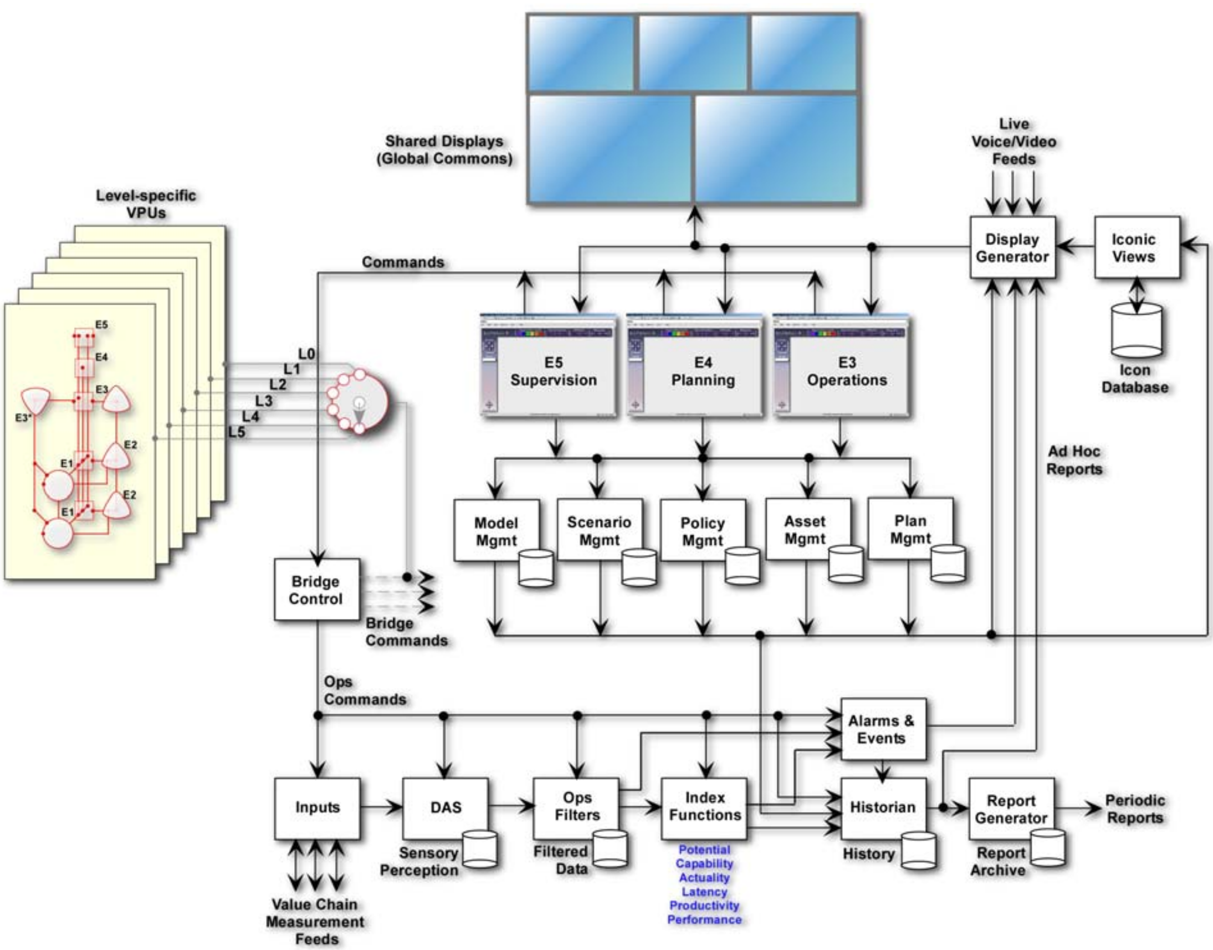

Figure 12 - EC2 Bridge Concept

The machinery supporting this collaborative environment is referred to as the "C2 Bridge." We expect that some form of bridge services are required in all enterprises that participate within the EC2 environment. Bridge services will also need to scale from some lowest common denominator (LCD – pun intended!) to sophisticated large-scale US Defense Department enterprise environments, as in Figure 11.

To support our model of EC2-guided VPUs, the bridge technical environment must contain elements introduced in Figure 6 and Figure 8. In particular, the C2 Bridge comprises a set of possibly shared services, as in an ASP[14], that support collaboration on both private (intra-VPU) and public (inter-VPU) situation assessment and behavior generation tasks. The bridge services are introduced in block diagram form in Figure 12. Situations inter the bridge at the lower left, are aggregated and filtered and entered first into a performance measurement process, then a historian. Some inputs qualify as alarms or important events and are presented to the Alarm & Event processor.

Current and historical information are subsequently input to a display generator that provides synoptic views on shared displays and actors' workstations. Based on this input actors respond through interaction with the core EC2 applications (A1-A7), as depicted in Figure 8. Actors also interact with the enterprise VPU through a level selection mechanism.

The bridge enables a wide range of control functions that support enterprise operations. The control functions and their impact on viability are the subject of another paper.

3 Conclusions

Our thesis is that:

> Effective (agile and error free) governance of large-scale distribute, always on, real-time enterprise that is capable of responding to probabilistic external and internal demands requires *systematic* (engineered) application of an intelligent decision and control system framework.

We have provided both a lexicon for expressing enterprise C2 and the outline for an operational framework for implementing scalable, distributed real-time processes of command and control. The work described here forms the basis for our current efforts at constructing an EC2 demonstration system, with special emphasis on the requirements of joint command and collaborative C2.

[14] ASP: an Internet service provision point, a web-based "Application Service Portal"

4 References

1. Alberts, D. and Hayes, R., Power to the Edge, Command and Control in the Information Age, DOD CCRP Publication, June 2003
2. Albus, J., Outline for a Theory of Intelligence, IEEE Transactions on Systems, Man and Cybernetics, Vol 21, No 3, May-June 1991
3. Alpur, D., et al, Core J2EE Patterns, Sun Microsystems Java 2 Platform Series, Prentice-Hall, 2001
4. Antsaklis, P., Passino, K., Wang, S., Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues, Journal of Intelligent and Robotic Systems, 1:315-342, 1989
5. Ashby, R., Introduction to Cybernetics, Chapman & Hall, Ltd, London, 1957, PDF version available by permission at www.echelon4.com
6. Bayne, J., A Software Architecture for Control of Value Production in Federated Systems, Presented at the World Multi-Conference on Systemics, Cybernetics & Informatics, Orlando, July 28th, and published in the Journal of Systemics, Cybernetics & Informatics, Vol. 1, No. 8, August 2003
7. Bayne, J., Architecture of Federated Enterprise, ISA 2003 Expo Conference, Houston, TX October 2003
8. Bayne, J., Automation & Control in Grid-Connected Systems, 3rd IEEE Electro-Information Technology Conference, Indianapolis, IN, June 2003
9. Bayne, J., Automation & Control in Large-scale Interactive Systems, Proc IEEE International Symposium on Object-Oriented Real-time Computer Systems (ISORC), 2002
10. Bayne, J., Distributed Real-time Enterprise, 2002, available at www.echelon4.com, a derivative of which has been submitted to the IEEE Transactions on Systems, Man and Cybernetics
11. Bayne, J., Feasibility of Grid-based Distributed Real-time Command & Control, Volume 1, Phase I Proposal, Topic AF03-094, Proposal No. F031-0494, January 6, 2003
12. Bayne, J., Feasibility of Grid-based Distributed Real-time Command & Control, Volume 2 Concept Specification V1.0, AFRL SBIR Contract F30602-03-C-0154, September 2003
13. Bayne, J., Feasibility of Grid-based Distributed Real-time Command & Control, Volume 3 Architecture Specification V1.1, AFRL SBIR Contract F30602-03-C-0154, October 2003
14. Bayne, J., Industrial Automation & Control in a Networked Economy, invited talk, IEEE Workshop on Object-oriented Real-time Distributed Systems (WORDS), Santa Barbara, CA, 1996
15. Bayne, J., MetaSystems, Proceedings of the Instrumentation, Systems, and Automation (ISA) Society, 1998
16. Bayne, J., Performance Measurement in C2 Systems, Proceedings of the 8th International Command and Control Research and Technology Symposium, National Defense University, Ft. McNair, Washington DC, June 2003
17. Bayne, J., Scalability of Performance in High Volume Commercial Control Systems, Proc IEEE International Symposium of Object-Oriented Real-time Systems (ISORC), 1999
18. Bayne, J., Seem, J., Drees, K., Adaptive Industrial Controls, 7th International Symposium on Artificial Intelligence in Real-time Control (AIRTC'98), October 1998, winner "Best Paper Award"

19. Bayne, J., The Architecture of Enterprise, an unpublished essay, available at www.echelon4.com, 2002
20. Bayne, J., The ELBA Object Model, Ada Runtime Environment Working Group (ARTWEG), ACM SIG Ada, Santa Barbara, CA, 1995
21. Beer, S., Brain of the Firm, 2nd Ed., Wiley, 1994
22. Booch, G., et al, The Unified Modeling Language, User Guide, Addison-Wesley, 1998
23. Brown, A., Johnson, S., Kelly, K., Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, Rational Software (IBM) Whitepaper, April, 2003
24. Buyya, R. and Murshed, M., GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, The Journal of Concurrency and Computation: Practice and Experience (CCPE), 1-32pp, Wiley Press, May 2002 (to appear)
25. Buyya, R., Economic-based Distributed Resource Management and Scheduling for Grid Computing, PhD Thesis, Monash University, Melbourne, Australia, April 12, 2002
26. Buyya, R., et al, Economic Models for Resource Management and Scheduling in Grid Computing, Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, May, 2002 (to appear)
27. C4ISR Architecture Framework, Version 2.0, US Department of Defense, 18 December 1997
28. Chinnici, R., et al, Web Services Definition Language (WSDL) - Version 1.2, WC3 Working Group Draft, July 2002, www.w3.org/2002/07/wsdl
29. Clark, R., Jensen, E., Wellings, A., Wells, D., The Distributed Real-time Specification for Java: A Status Report, www.real-time.org, Revised January 2002
30. Copeland, T., Koller, T., Murrin, J., Valuation – Measuring and Managing the Value of Companies, 3rd Ed., Wiley, 2000
31. Damianou, N., Dulay, N., Lupu, E. and Sloman, M., The Ponder Policy Specification Language, Lecture Notes in Computer Science, vol. 1995, 2001
32. De Roure, D., et al, Evolution of the Grid (pre print), Universities of Portsmouth and Southampton, UK. In Grid Computing: Making the Global Infrastructure a Reality
33. Deering, S., Hinden, R., Internet Protocol, Version 6 (IPv6) Specification, Network Working Group, Internet Engineering Task Force (IETF), December 1998, ftp://ftp.isi.edu/in-notes/rfc2460.txt
34. Defense Information Systems Agency (DISA), Strategic Communications Office, Unified Command Structure (UCS), Architecture Development – Overview and Summary Information (AV-1), June 9, 2003
35. Defense Planning Guidance (DPG), May 2002
36. Department of Defense Architecture Framework v1.0
37. DOD, C4ISR Architecture Framework - Version 2.0, December 1997
38. DOD, E-6 Command, Control, Communications, & Intelligence ACTD Proposals, September 2001
39. DOD, Joint Tactical Architecture – Version 4.0, April 2001
40. DOD, Office of Force Transformation, Network Centric Operations Conceptual Framework, Draft Version 1.0, August 29th, 2003
41. Foster, I., et al, Grid Services for Distributed System Integration, IEEE Computer, June 2002
42. Foster, I., et al, The Physiology of the Grid – An Open Grid Services Architecture for Distributed Systems Integration (DRAFT), Argonne National Lab, June 2002

43. Foster, I., et al, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration (Draft 2.9), June 2002
44. Foster, I., The Grid: A New Infrastructure for 21st Century Science (Reprint), Physics Today http://www.aip.org/pt/vol-55/iss-2/p42.html
45. Galiasso, P., Bremer, O., Hale, J. and Shenoi, S., Policy Mediation for Multi-Enterprise Environments, In Proc. 16th Annual Computer Security Applications Conference, 2000
46. Gamma, E., et al, Design Patterns, Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
47. Global Information Grid (GIG) Enterprise Service, Initial Capabilities Draft, 11 June 2003
48. Halang, W. and Stoyenko, A., Ed., Real Time Computing, NATO ASI Series F: Computer and Systems Sciences, Vol 127, 1991
49. Harry, M. and Schroeder, R., Six Sigma, Currency (Doubleday), 2000
50. Henoch, J. and Ulrich, H., Agent-based Simulation Platform for Evaluating Management Concepts, Proceedings of EUROSIM 2001, June 2001
51. Herring, C. and Kaplan, S., The Viable System Model for Software, Department of Computer Science and Electrical Engineering, University of Queensland, Brisbane, Queensland, QLD 4072, June 2000
52. House of Representatives Report 107-532 on the DoD Appropriations Bill 2003, Data Management, 15 March 2003
53. Jaques, E., Requisite Organization – The CEO's Guide to Creative Structure & Leadership, Cason Hall, 1989
54. Jensen, E., Locke, C., and Tokuda, H., A Time-Value Driven Scheduling Model for Real-time Operating Systems, Proc. Symposium on Real-time Systems, IEEE, November 1985
55. Jensen, E., The Distributed Real-Time Specification for Java – An Initial Proposal, Journal of Computer Systems Science & Engineering, March 2001
56. Jensen, E., Utility Functions: A General Scalable Technology for Software Execution Timeliness as a Quality of Service, Proc. Software Technology Conf., Utah State Univ., April 2000
57. Johnson, W. and Brooke, J, Core Grid Functions: A Minimal Architecture for Grids, Working Draft 3, Grid Protocol Architecture Working Group, Global Grid Forum, July 2002
58. Joint Command and Control Functional Concept, 5 September 2003
59. Joint Command and Control Functional Concept, Agile C2, 10 July 2003
60. Joint Force Command and Control Concept to Guide Standing Joint Force Headquarters Development by 2005, 23 January 2003
61. Joint Operations Concept, 19 August 2003
62. Kangasluoma, M., Policy Specification Languages, http://www.nixu.fi/~minna/draft2.html, 1999
63. Kishimoto, H. and Snelling, D., OSGA Fundamental Services: Requirements for Commercial Grid Systems, OSGA Working Group, Global Grid Forum, October 2002
64. Lenahan, J., An Abstract Model for Evaluating Unified Command and Control, A Scenario and Technology Agnostic Approach, Draft V1.0, OASD Internal Document, October 9, 2003
65. Lessig, L., Code, and Other Laws of Cyberspace, Basic Books, 1999; see also www.cyberlaw.stanford.edu/lessig/
66. Locasto, M., Burnside, M., Li, C. and Wahl, A., SPCL: Structured Policy Command Language, http://www1.cs.columbia.edu/~locasto/projects/spcl/docs/whitepaper.html, 2003
67. McKinsey, J., Introduction to the Theory of Games, McGraw-Hill, 1952

68. Moffet, J., Complexity Theory, CCRP Publication Series, September 2003
69. Morrison, J.G., et al, The Knowledge Wall for the Global 2000 War Game: Design Solutions to Match JOC User Requirements, Technical Report 1860, SPAWAR San Diego, August 2001
70. Nuclear Command and Control System End to End Review (ETER), April 2002
71. Nuclear Posture Review (NPR), December 2001
72. Object Management Group (OMG), UML Profile for Schedulability, Performance, and Time Specification, Final Adopted Specification, March 2002
73. Paul, R., Performance Evaluation and Monitoring Using Scenario-Driven System Engineering (SDSE) for Command and Control, PowerPoint Presentation, OASD/NII, November, 2003
74. Pfleeger, C. and Pfleeger, S., Security in Computing, 3rd Edition, Prentice Hall PTR, 2003
75. Pouchard, L., et al, Ontology Engineering for Distributed Collaboration in Manufacturing, Proceedings of the AIS2000 Conference, March 2000
76. Quadrennial Defense Review Report (QDR), 30 September 2001
77. Quatrani, T., Visual Modeling with Rational Rose 2002 and UML, Addison-Wesley, 2003
78. Rajic, H. et al, Distributed Resource Management Application API Specification, Global Grid Forum, September 2002
79. Rumbaugh, J., et al, Object-oriented Modeling and Design, Prentice-Hall, 1991
80. Ryutov, T. and Neuman, C., Representation and Evaluation of Security Policies for Distributed System Services, In Proc. DARPA Information Survivability Conference Exposition, 2000
81. Schlenoff, C., et al., The Process Specification Language (PSL), Overview and Version 1.0 Specification, NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD (2000)
82. Schwiegelshohn, U. and Yahyapour, R., Grid Scheduling Architecture, Working Draft 1, Global Grid Forum, October 2002
83. Scruddrer, R., Lutz, R., Dahmann, J., Automation of HLA Federation Development and Execution Process, williams.af.mil/ambr3.htm
84. Selig, B., Gullekson, G., Ward, P., Real-time Object-Oriented Modeling, Wiley, 1994
85. Smith, W., A Framework for Control and Observation in Distributed Environments, NAS Technical Report #NAS-01-006, June 2001
86. Sun Microsystems, Java Remote Method Invocation Specification, December 1999
87. The National Security Strategy of the United States of America, September 2002
88. The National Strategy for Combating Terrorism, February 2003
89. The National Strategy to Combat weapons of Mass Destruction, December 2002
90. Transformation Planning Guidance (TPG), US Department of Defense, April 2003
91. Tsai, WT. and Paul, R, Adaptive, Scenario-based, Object-oriented Test Frameworks for Testing Embedded Systems, ITEA Journal, June/July 2003
92. Tuecke, S., et al, Open Grid Services Infrastructure (OGSI), Version 1.0, GFD-R-P.15 (Proposed Recommendation), June, 2003
93. UCS Project and UCS Roadmap, Documents and Products Plan, OASD/NII Internal Document, DRAFT V1.4, September 30, 2003
94. Unified Command Plan (UCP), 30 April 2002, including Changes 1 (30 July 2002) and 2 (10 January 2003)

95. Unified Command Structure (UCS) Operational Concept V1.11, OASD/NII Draft, October 27, 2003
96. USAF Scientific Advisory Board, "Building the Joint Battlespace Infosphere Volume 1: Summary" SAB-TR-99-02, http://www.rl.af.mil/programs/jbi/documents/JBIVolume1.pdf, 1999
97. Whitman, L. and Huff, B., The Living Enterprise Model, Automation & Robotics Research Institute, University of Texas at Arlington, 2000
98. Wooldridge, M., Reasoning About Rational Agents, MIT Press, 2000