A Theory of Software-Mediated Enterprise Governance

Jay S. Bayne
Meta Command Systems, Inc.
jay.bayne@metacomsys.com

Abstract

Cybernetics, among its more general definitions, is about effective governance of dynamic probabilistic systems. First order cybernetics is concerned with [software] processes governing behavior of such systems. Second order cybernetics is concerned with [software] process defining the context, including observers and actors, within which behaviors of these systems unfold. Together, they define an important branch of the emerging field of software cybernetics, a branch concerned with the theory of software-mediated governance systems and its supporting theories of enterprise command and control (EC2). This paper outlines key elements of an EC2 theory.

Keywords: *cybernetics, command and control (C2), enterprise systems*

1. Introduction

Our theory of *enterprise command (decision) and control* (EC2)[1] is concerned with effective governance systems supporting *value production* in distributed *intelligent enterprises*[2], about increased institutional awareness and its resultant ability to facilitate more effective, responsive (agile) and sustainable (continuous and valid) unilateral and multilateral action. Additionally, EC2 is about improved integration of the prerogatives of human actors and distributed computational systems that, acting in concert, are competent to establish and maintain viability[3] of large-scale, dynamic and increasingly complex enterprise systems.

The theory maintains that intelligent enterprises are not possible without a more formal EC2 computational infrastructure, one based on a *unified* command and control *framework*. Such an infrastructure logically sits above yet depends upon such underlying information services as *enterprise resource planning* (ERP), networks such as the Internet, corporate intranets, the DoD's *global information grid* (GIG) and attendant *network-centric enterprise service*s (SOA/NCES). To this end, the theory of EC2 offers a cybernetic[4] framework for enabling collaborative, distributed and time-critical *service-centric operations*.

Our thesis is that the definition, development and deployment of interoperable service-centric EC2 systems require a more formal degree of specification and standardization than present in *ad hoc* and essentially social governance systems. In response, we have developed an EC2 framework comprising six key elements: i) a dynamic enterprise model, ii) a command (actor) model, iii) a set of supporting command services, iv) a set of control services, v) a set of scale-free performance measurement services and vi) interfaces to legacy operational software systems. In this limited paper, we summarize the enterprise (i) and services-oriented software command (iii) and control (iv) models.

Our theory postulates that without higher and more robust forms of enterprise automation and control, current institutions of government, commerce and society will be progressively challenged (constrained) in their individual and collective actions[5], especially in an increasingly complex, interdependent and interactive world. The goal of a successful theory of EC2 is to move the frontier of enterprise systems conception, design, deployment and operation from its current communications (net-centric) and information sharing (*data-centric*) orientations forward to theories and software engineering practices based on coordinated (*service-centric*) operations.

Today, for example, both commercial and government enterprises emphasize communications

[1] This paper summarizes key elements of a more general theory described in the author's [to be published] text *Theory of Enterprise Command and Control*, October of 2005 [1].

[2] The term *enterprise* refers to an arbitrary unit of organization accountable for sustainable production of a quantifiable measure of value.

[3] An enterprise is *viable* to the degree it maintains its value propositions (its capabilities) over time and within the federated contexts in which it operates.

[4] Core specifications of cybernetic systems include specification of their dynamics, policies for establishing and maintaining their homeostasis around one or more value propositions and their recursive governance [feedback control] mechanisms.

[5] Witness, for example, difficulties encountered in individual and collective responses to natural (e.g., Katrina) and terrorist (e.g., 09.11.01) disasters and failures in major private sector corporations such as Enron, WorldCom, HealthSouth and Tyco.

(network transport) and data centricity (information sharing) as cornerstones of their integration efforts. The ability to interconnect arbitrarily large numbers of distributed and diverse information sources is a logical necessity, but as a basis for unified governance, is insufficient. Large quantities of volatile context-sensitive uncorrelated information streaming in (whether pushed or pulled) from various geographically distributed sources are as likely to confuse a manager's (commander's) assessment of a given situation as to clarify it, clouding his/her ability to act.

Moreover, there are important collateral issues of information pedigree, timeliness (completion-time requirements), precision, semantics (ontology) and a host of other matters that connectivity and data access alone cannot resolve. These issues properly belong to the domain of the governance *processes* (i.e., net-centric services) that ultimately consume, interpret and utilize (act upon) the data in support of human prerogatives. These issues define the environment (observer *context*) in which EC2 processes unfold in time and space (cyberspace-time) and the missions of organizations that rely on results they provide.

In short, we believe that communications networks and their ability to support an abundance of data publishers and subscribers play a critical but supporting role in enabling effective and collaborative enterprise C2. This perspective motivates the next developmental phases of institutional (public and private enterprise) governance systems. It argues that the focus needs to move to the *assimilation* and effective *utilization* of information within and among communities of allied enterprises. This is the domain of collaborative real-time governance, of institutional and collective awareness, and essential software-enabled *processes* of enterprise C2. It is the domain addressed by our general theory of EC2.

2. Objectives

A key objective of this paper is to outline three primary elements of a broadly applicable (i.e., domain neutral, logical and scalable) *unified theory of enterprise command and control* applicable to the management of multi-domestic commercial enterprises, asymmetric network-centric warfare and homeland defense in response to natural and terrorist disasters.

A collateral objective is the creation of an EC2 reference model (EC2/RM) for software-mediated *collaborative real-time governance* of an enterprise. A *real-time enterprise* is one whose essential governance activities, within probabilistic bounds, achieve specific completion-time requirements for its own internal (unilateral) and collaborative external (multilateral) activities. This feature motivates and distinguishes our approach from more traditional non-real-time (i.e. *ad hoc*) C2 systems. More specifically, a core requirement for what follows is the management of end-to-end timeliness of decision and control actions, especially in situations where collaboration and synchronized action are critical.

3. Enterprise – The Object of Governance

EC2 theory defines enterprise command and control, be it military or civilian, governmental or commercial, public or private, as *governance of the processes of value production*[6]. Mechanisms of value production are encapsulated in an enterprise. They are the *object* or focus of EC2's attention. An enterprise is an abstract or virtual machine[7] referred to in our theory as a v*alue production unit* (VPU). A VPU is a service-bearing object encapsulating one or more well-defined and manageable *capabilities*[8].

Our VPU is *mesosynchronous*[9] and operates in three dimensions. As diagrammed in Fig. 1, the first two dimensions of governance services function continuously at the intersection of a vertical *command axis* (aka, accountability or superior-subordinate network or asset chain) and horizontal *production axis* (aka, effects, peer-peer or production network or supply chain). The flow of information defining orders, resources, events, situations, policies, plans and effects on each axis intersect at the VPU's *management core*[10], requiring competent information fusion, analysis, proactive and reactive planning, risk assessment, compromise and decision among time dependent and typically conflicting operational demands. This is the essence of governance, where value production emerges from the process of decision and control. Table 1 summarizes the primary function of each interface on the two primary axes of a VPU.

[6] *Value* is defined in terms of specific *value propositions* associated with enterprise missions, goals and objectives.

[7] A *virtual machine* is any non-physical construct that functions (operates) within the confines of another real (physical) or virtual machine

[8] A *capability* is a quantifiable means of accomplishing a specific task.

[9] A *mesosynchronous* system operates in the middle ground between purely synchronous and purely asynchronous timing considerations.

[10] The *management core* is represented physically by a "C2 enclave" or "C2 bridge" where commander and officers gather to govern an enterprise and its mission(s). By analogy, an enterprise is a *ship*, the command staff is the *flight crew* and the C2 enclave is the *flight deck*.

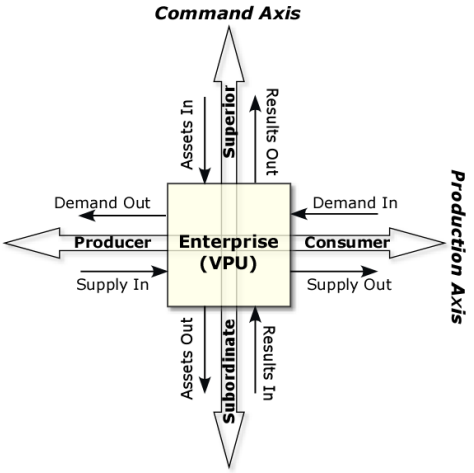

Fig. 1 – Enterprise Value Production Unit

Fig. 2 introduces a third dimension, showing a VPU along with its immediate command and production axis neighbors. It also introduces a naming (indexing) scheme that supports the requirement that each enterprise be uniquely identifiable. The focus of the figure is the central enterprise designated $VPU_{j,k,l}$. Index "j" identifies the operational domain (i.e., federation, or community of interest), "k" denotes the horizontal position in the federation's production network and "l" represents the location in its vertical command network. Within a given context the enterprise VPU has a single *superior* designated as $VPU_{j,k,l+1}$ and potentially many *suppliers* designated as $VPU_{j,k-1,l}$, *customers* designated as $VPU_{j,k+1,l}$ and *subordinates* designated as $VPU_{j,k,l-1}$. Notice in the figure that each supplier, customer and subordinate VPU may actually represent (i.e., be a proxy for) multiple active neighbors. This detail is important technically, but we will not labor on its relevance further.

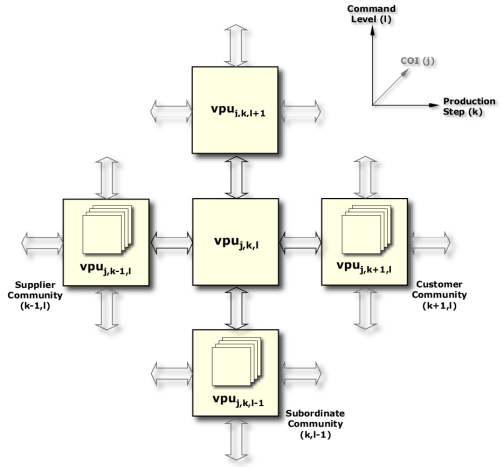

Fig. 2 – Enterprise (VPU) Structure

Enterprises typically operate in several *federations*, *contexts* or *communities of interest* (COI). Federations are alliances formed by members striving to develop and exploit, for individual and mutual benefit, higher-order[11] value propositions. The role of VPU governance in balancing these self- and community-serving value propositions is critical to the viability of both individual enterprises and the federations in which they operate.

Table 1 - Enterprise VPU Interfaces

Axes	Port Name	Port Function
Command Axis	Assets In	Acceptance and assimilation, according to service-level agreements (SLA), of allocated assets and tasking orders from superior VPUs
	Returns Out	Production of returns on value produced by previously allocated assets or issued commands; requests for allocation of additional assets; clarification requests on issued tasking orders
	Assets Out	Issuance of assets and tasking orders to subordinate VPUs with expectations for a time-bound returns of value produced
	Returns In	Acceptance and assimilation of returns and receipt and evaluation of requests for new asset allocations or readiness for new commands from subordinate VPUs
Production Axis	Demand In	Acceptance and assimilation, according to SLA, of demand orders for goods or services from upstream consumer (client) VPUs
	Supply Out	Fulfillment of previously received demand orders to downstream consumer (client) VPUs
	Demand Out	Issuance of demand orders for goods or services to upstream producer (server) VPUs
	Supply In	Receipt and acceptance of fulfillment related to previously issued demand orders for goods or services from upstream stream producer (server) VPUs

According to Jeffersonian principles, the so-called axioms of free society[12], federation members are semi-autonomous and self-regulating. Their designs are required to allow them to be 1) viable and uniquely identifiable members of one or more federations, 2) governed by federation laws (policies), and 3) to provide their individual contributions (capabilities) to the coherent ensemble behaviors that characterize missions, goals and objectives of the federated enterprise as a whole.

[11] A *higher-order value proposition* is one that benefits but transcends any given member's objectives, requiring capabilities formed from member asset sharing and collaborative C2.

[12] http://www.lewrockwell.com/vance/vance17.html

Membership in multiple federations effectively defines a third (z-axis) dimension to the VPU model. Fig. 3 amplifies the model in Fig. 2, defining a VPU's operating context, referred to as "enterprise C2 space." The central enterprise is uniquely identified as $VPU_{j,k,l}$. As depicted, it holds membership in four COI, numbered j=1 through j=4. In order for the central VPU's management team to maintain its commitments (i.e., realize its value propositions) within each COI, its EC2 governance function must provide unambiguous (context-sensitive) command and control over activities within and across each context.

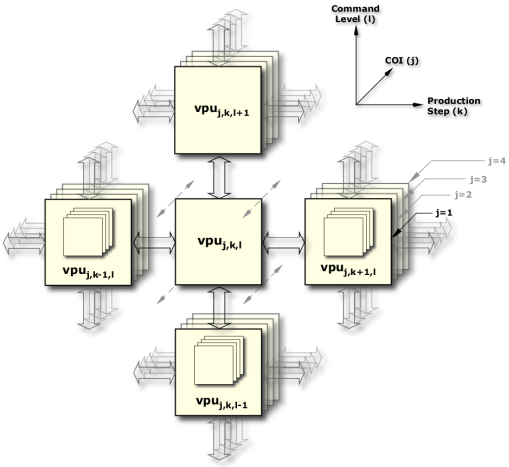

Fig. 3 – Enterprise C2 Space

Note the required multidimensional, computational multitasking and real-time nature of EC2 processes and the corresponding requirement for a VPU to move in an agile manner among (*context switch* between) different federations and activities. To do so requires that VPUs exhibit the ability to save the current state of each COI engagement when its attention must switch to a competing COI task, enabling it to return and correctly pick up saved *execution threads* in a time-bound, information lossless and coordinated manner. Furthermore, the C2 enclave (flight deck) where VPU command actors (flight crew) work must maintain appropriate views of these various contexts, their priorities, conflicts and interdependencies.

One of the more important context-sensitive issues for command to monitor in an enclave is the set of federation rules (policies) that constrain its prerogatives. Enterprises and the federations to which they belong may operate under a wide range of policies (rules of engagement, civil laws, military doctrines, etc.). Policies relevant to one COI may not apply to, or indeed may conflict with, policies relevant to another. In such cases, policy awareness through validation mechanisms must be a formal part of standardized EC2 services.

4. Control Processing Services

Fig. 4 introduces three primary enterprise control-processing stages supporting the enterprise regulatory feedback loop: situation assessment (SAS), plan generation (PGS) and plan execution (PES). This formulation is generalized and meant to convey the core functions required of an EC2 theory. Other compatible formulations are possible[13]. The figure identifies seven key processing steps in moving from raw sensor-based observations to executing plans that affect actuators and drive the process to some desired next state.

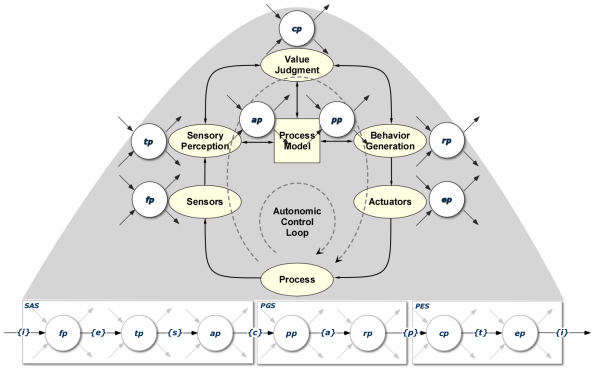

Fig. 4 – Regulatory Control Processing Services

Small circular symbols used to represent each step contain multiple arrows into and out of their respective services. Inbound arrows represent their subscriptions to a VPU's various internal and external publishers of relevant information topics. Outbound arrows represent a VPU's publications to its internal and external topic subscribers. The implication is that byproducts of individual control service steps within a given VPU may support the C2 processing of other allied VPUs. While the primary role of a VPU's control services is guiding its own self-directed behaviors, it may simultaneously use its capabilities to assist in the governance of other VPUs within the larger federated enterprise. Alternatively, it may desire or require the specialized services of other command or production chain affiliates to accomplish its internal objectives. This view is consistent with and complementary to the asynchronous end-end communications strategy referred to as "task, publish, process then use" (TPPU).

[13] See, for example, Boyd's OODA loop [3], Lawson's C2 Process Model [4], Wohl's SHOR model [5], JOPES [6] and MAPPER [7].

4.1. Situation Assessment Services (SAS)

Situation assessment services include processes of observation (measurement), awareness (recognition) and analysis (understanding). SAS provides the front-end to the remaining EC2 services. The lateral flow of information through the SAS stage passes through three primary functions. A *filter process* searches information (e.g., subscriptions) *{i}* for occurrence of significant events *{e}*. A *triage process* subsequently correlates these event lists in order to recognize new or changes in situations *{s}*. Finally, an *analysis process* converts new or evolving situations into candidate responses in the form of scenarios or courses of action (COA) *{c}*.

4.2. Plan Generation Services (PGS)

There are two primary plan generation services, providing policy compliance and resource reservations for proposed courses of action *{c}*. COA enter the PGS stage where they are analyzed by a *policy process (pp)* that, utilizing a context-specific *policy*[14] database, produces zero or more policy-compliant plans of action (POA) *{a}*. Then, utilizing an asset or *resource*[15] database containing the status of available resources, the *resource process (rp)* converts the POA into zero or more (e.g., a primary and back up or alternate) executable plans of record (POR) *{p}*.

4.3. Plan Execution Services (PES)

Plan execution services of a VPU include two primary processes. Policy-validated and resource-allocated plans of record *{p}* enter at the left in Fig. 4, arriving first at the *command process (cp)*. Here they are reviewed, scheduled and authorized by the VPU's senior officer, emerging in the form of *tasking orders {t}*. Tasking orders are subsequently dispatched to the *execution process (ep)* for assignment, activation, execution, synchronization and continuous monitoring.

Supervising these seven control processing services are EC2 actors, typically the human managers or their agents responsible for the conduct of the enterprise. The services provided to these actors are referred to as command processing services. Their definition follows now classical management cybernetic principles.

[14] A *policy* is a set of [if…then…else…] conditions governing actions (i.e., use of specific capabilities) within a given *policy domain.*

[15] A *resource* is any consumable or serially reusable asset (men or material) that may be used in "resourcing" a capability, often managed by ERP applications..

5. Command Processing Services

The enterprise *command structure*, diagrammed in Fig. 5 and enumerated in Table 2, derives from five decades of cybernetics [8, 9], systems theory [10], operational research [11] and theories of human consciousness and neuroanatomy [12]. Its role in modeling consciousness and adaptive behavior is documented in [13]. The figure shows relationships among the principal VPU actors responsible for the conduct of enterprise operations.

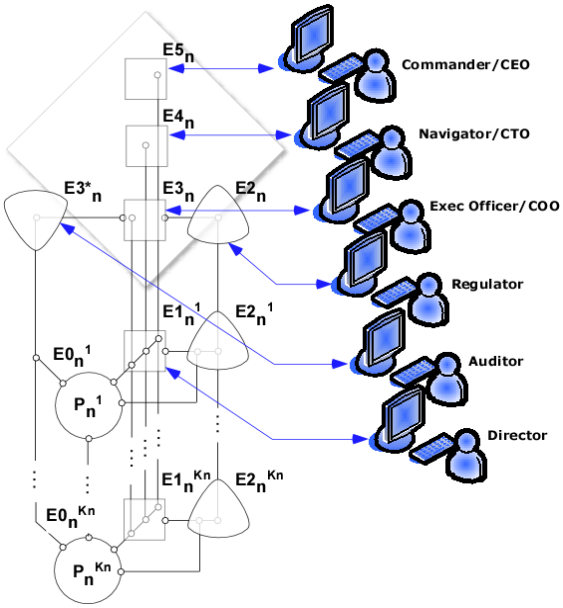

Fig. 5 – EC2 Command Structure

Cybernetic governance structures are fractal, exhibiting recursive self-similar structures [14]. Each VPU contains a command structure capable of self-governance. Command structures, typically co-located in mobile or fixed command "enclaves," are actor teams that operate semi-independently with a high degree of autonomy and self-determinism.

Principal per-VPU actors include:

- A single *commander* or command staff (aka, CEO, supervisor, director, or manager, denoted as echelon five, E5) representing the highest authority within the VPU
- A single *navigator* or planning staff (aka, planner-analyst, denoted as echelon four, E4) responsible for modeling, planning and analysis functions (e.g., adaptation and change management)
- A single *operator* or operations staff (aka, COO, operations executive, denoted as echelon three, E3)

responsible for the execution of authorized plans of record

Table 2 - Principle EC2 Actors

Label	Services	Roles & Responsibilities
E5	Command	Mission Goals & Objectives, Policy & Command Authority
E4	Analysis/Planning	Modeling, Situation Assessment & Plan Generation
E3	Operations	Plan Execution & Capability Management
E3*	Audit	Program & Process Performance Assessment
E2	Regulation	Plan (Task) & Resource Synchronization
E1	Direction	Plan (Task) Execution Management
E0/P	Process	Embedded [Value] Production Process

Fig. 6 shows a sequence of subordinated (embedded) enterprises, each governing successively lower levels of the Level "n" enterprise. At each level, the three primary actors (E5, E4 and E3) are contained in a rectangle (rotated 45°) to emphasize both their leadership role and the nested or recursive nature of their position within the ECS structure.

The figure shows the Level "n" organizational unit at the left. The central figure shows the Level "n" subordinate enterprises at Level "n-1" (i.e., P_n^1 through P_n^K), with each symmetrically defined and rotated 45°, as are their respective Level "n-2" structures diagrammed at the right. This sequential nesting is a key feature of the ECS model within our theory of C2. Command concepts are coherent, structured and uniformly applicable throughout the enterprise accountability hierarchy. This recursive property is the foundation for accountability and the collaboration framework necessary to reinforce it.

As noted in Fig. 5 and 6, a given VPU may contain "K_n" subordinate processes within its authority at Level "n," labeled P_n^1 through P_n^{Kn}. In the EC2 model outlined here, the minimum number of such processes is two (K_n=2), one for value production along the command axis and one governing value production along the production axis.

In addition to and in direct support of the three principal EC2 actors, there are:

- Two or more subordinate *directors* (defined at echelon one, E1) of the internal functional capabilities (embedded VPUs, at least one for the asset chain, and one for the supply chain)
- *Regulators* (defined at echelon two, E2) responsible for the synchronization of subordinate VPUs in their execution of coordinated tasks that must rendezvous in time or synchronize on shared serially-reusable resources (providing "excitatory" or stimulus controls)
- An *auditor* (defined at echelon three star, E3*) responsible to E3 for continuously measuring and reporting on the performance of subordinate VPUs (providing "inhibitory" or damping controls)
- Two or more subordinate value *production processes* (denoted as echelon zero, E0 or P_n) are managed by their respective E1 actors

The control loop defined on the right side of Fig. 5, labeled E3-E2-E1-E3, provides the *excitation pathway*[16] responsible for initiation and amplification of activities in subordinated processes. The loop on the left side, labeled E3-E3*-E0-E1-E3, is the *inhibition pathway*[17] responsible for restraining or attenuating activities in subordinated entities.

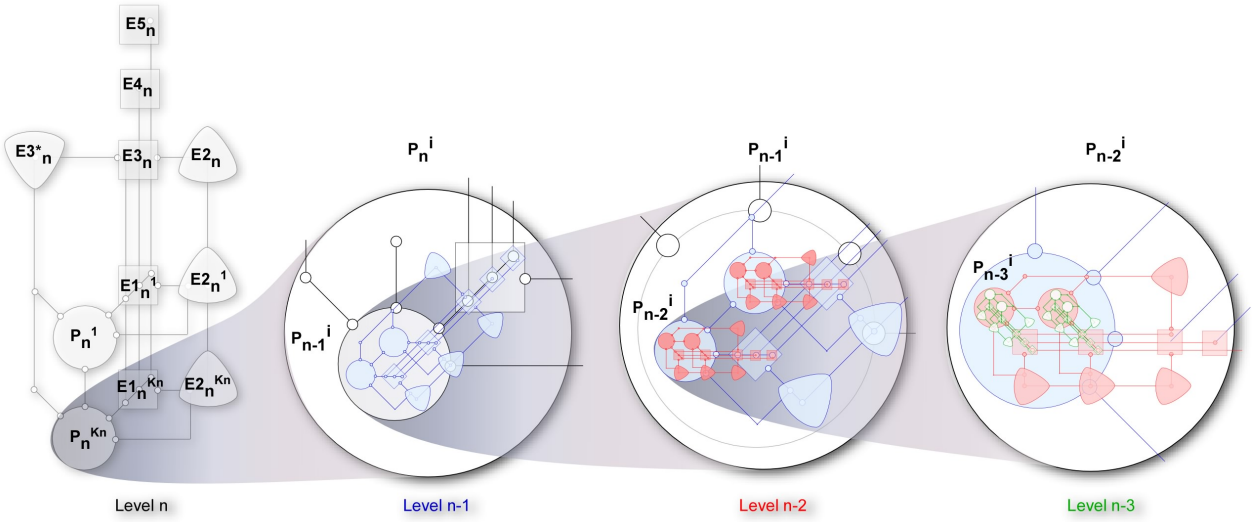

Fig. 6 – Command Axis Recursion (nested accountability hierarchy)

This approach to defining EC2 capabilities offers several benefits. First, the recursion creates a structure whereby applications (software-based C2 services) designed for one level of command can, in principle, be deployed at levels above and below it in the command hierarchy. Second, a commander trained to use EC2 services a one level can ascend (descend) to the next level of command and be at home with the concepts and mechanisms of C2 at that level. Third, EC2 systems whose implementations are validated at a given level of command are in principle valid at other levels. Fourth, documentation and training prepared for a given level are relevant at other levels with minor adjustments.

6. Epilog

This paper is necessarily brief. Several recent publications present related aspects of the EC2 theory introduced here. [15] discusses enterprise *performance measurement services* (PMS), [16] outlines elements of the VPU software *object model*, [17] presents elements of *policy management* in the real-time regulation of enterprise behavior, [18] discusses the *scale free* nature of the command model and [19] applies the theory to the design of C2 systems supporting the DoD's *Joint Task Force* (JTF) concept.

7. Acknowledgments

Elements of work reported here were funded by the Air Force Research Laboratory, Rome, NY under contracts F30602-03-C-0154 and FA8750-04-C-0084.

8. References

http://www.echelon4.com/references.htm provides a more comprehensive list of references related to the subject of EC2 requirements and theory, including those references listed below.

[1] Bayne, J., Theory of Enterprise Command and Control, an unpublished text, pp 220, October 2005

[2] Whitman, L. and Huff, B., The *Living Enterprise Model*, Automation & Robotics Research Institute, University of Texas at Arlington, 2000

[3] Brehmer, B., *The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control – Assessment, Tools and Metrics*, 10th International Command and Control Research and Technology Symposium, June 2005

[4] Lawson, J., *Command and Control as a Process*, IEEE Control Systems Magazine, March 1981

[5] Wohl, J., *Force Management Decision Requirements for Air Force Tactical Command and Control*, IEEE Transactions on Systems, Man and Cybernetics, Vol 11, No 9, 1981

[6] DOD, Joint Operations Concept (JOC), 19 August 2003

[7] DOD, US Joint Forces Command (USJFCOM), *Operational Concept for Global/Joint Battle Management Command and Control (GJBMC2) Working Paper*, Draft Version 9N, Bedford

[8] Weiner, N., Cybernetics – Communications and Control in Man and Machine, MIT Press, 1948

[9] Ashby, R., Introduction to Cybernetics, Chapman & Hall, London, 1957

[10] Albus, J., *Outline for a Theory of Intelligence*, IEEE Transactions on Systems, Man and Cybernetics, Vol 21, No 3, May-June, 1991

[11] Beer, S., Decision and Control – The Meaning of Operational Research and Management Cybernetics, Wiley, 1988

[12] Edelman, G. & Tononi, G., The Universe of Consciousness, Basic Books, 2000

[13] Bayne, J., *MetaSystems*, Proceedings of the Instrumentation, Systems and Automation (ISA) Society, 1998

[14] Beer, S., The Brain of the Firm, 2nd Edition, Wiley, 1994

[15] Bayne, J., *Performance Measurement in C2 Systems*, Proceedings of the 8th ICCRTS, National Defense University, Ft. McNair, Washington DC, June 2003

[16] 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

[17] Bayne, J. and Paul, R., *Policy Based Command and Control*, 10th ICCRTS, June 2005 (Best Paper nomination)

[18] Bayne, J. and Paul, R., *Scale Free Command and Control*, 10th ICCRTS, June 2005

[19] Bayne, J. and Diggs, D., *C2 in the Joint Task Force Enterprise*, 11th CCRTS, June 2006 (Best Paper nomination, to be published)

