Distributed Governance in Federated Systems

Jay Bayne, Senior Member IEEE & Distinguished Engineer ACM

Abstract **The coordination of interactive intelligent automation systems, including systems built over time from components provided by different suppliers, requires a shared design framework competent to express policies and mechanisms for guiding ensemble behaviors. We introduce here a governance framework that addresses recurring problems found in ad hoc solutions to systems integration, operations and maintenance challenges. Architectural frameworks are intended to provide guidance, as expressed through design patterns, which enable scalability, resilience and reusability of a set of core services, including support for discovery, collaboration, timeliness and fault management. Our thesis is that designing for adaptability and predictable performance in internetworked intelligent automation systems requires a consistent and coherent notion of distributed governance. Our framework supports a science of cyberspatial mechanics – space-time considerations governing dynamic behaviors of individual and federated assemblies of intelligent cyberspatial objects.**

Keywords—Cyberspace, cybernetics, cyberphysical systems, distributed control, enterprise governance, intelligent automation, service systems

I. Introduction

We are interested in the development of intelligent *service systems* [18] and resolution of associated design challenges incurred in managing their interactions. Their individual and collective behaviors are seen to unfold in *cyberspace*, an abstract 3-dimensional operating environment circumscribed by *geospatial*, *infospatial* and *sociospatial* extents. As previously introduced [4], *cyberspace* is an operational domain populated by interactive service-oriented [19] *cyberspatial objects* (CSO) [9], intelligent automation whose purpose and enabling capabilities are defined in terms of the quantifiable *value propositions* they offer each other. In the lexicon of the day, CSO dedicated to the regulation of physical processes are referred to as *cyberphysical systems* (CPS). The principle operational imperatives of CSO include processes for governing the production of goods or services offered to other CSO, typically resulting in formation of formal and informal *supply chains*. In addition to maintaining their unique value propositions, the *viability* of CSO operating as productive intelligent objects depends on their self-awareness, self-reliance and ability to adapt in the face of failures and competition for resources – characteristics that require detection and effective response to a variety of environmental pressures arising within the ecosystems in which they operate.

Lessons derived from six decades of applied *cybernetics* [1, 3, 6, 7, 10, 20, and 21] inform our understanding of the laws of governance found in natural and synthetic systems [5]. As such, synthetic CSO are deemed viable to the degree their value propositions remain viable over time, requiring adaptation while maintaining dynamic stability (homeostasis) as they provide services under probabilistic and competitive timing and resource demands. Achieving homeostasis requires CSO to exhibit agile internal governance processes (e.g., situational awareness, decision and control) competent to simultaneously assimilate information flows on supply chains coupling their clients (consumers) and their suppliers (producers), on *asset chains* coupling their superiors (parents) and their subordinates (children), and on a variety of sensors that measure the status of relevant external and internal resources. Consequently, CSO operating alone or in concert require a reliable internal feedback control mechanism, ideally one shared among collaborating partners – in short, a distributed federation governance system through which stability, adaptation and cooperation can be achieved.

II. Cyberspatial Objects

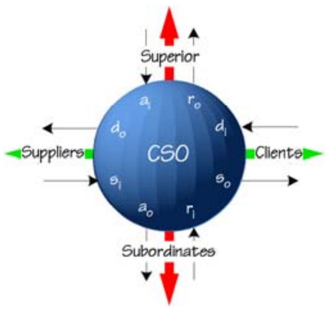


Figure 1 - Cyberspatial Object

Cyberspatial objects (Fig. 1) are modeled as abstract service providers defined by the value propositions they offer other CSO through *service access ports*, portals identified by a unique set of cyberspatial addresses. As depicted, along their horizontal axis CSO support connections to one or more supply chains. On this axis CSO serve their clients through trading protocols governing the receipt of demands ("orders") on demands_in ports (d_i) and the issuance of product or service responses on supplies_out ports (s_o). Likewise, CSO place their demand orders to suppliers on demands_out ports (d_o), subsequently receiving supplier responses on supplies_in ports (s_i).

Along vertical axes asset chains CSO, connected by command and control protocols, engage in communications with their superiors and subordinates. CSO receive commands (e.g., financial assets and policies) from superiors on their assets_in ports (a_i) and provide results (e.g., returns on assets and command responses) on their returns_out ports (r_o). Likewise, CSO place their own demands on subordinates on their assets_out ports (a_o) and receive results on their returns_in ports (r_i).

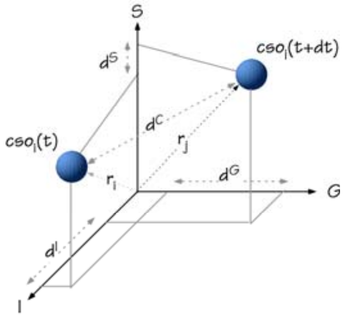


Figure 2 - Dynamics

CSO operate in a virtual world bounded by geospatial (G), infospatial (I) and sociospatial (S) dimensions (Fig. 2). Geospace defines a CSO's physical location, typically expressed in earth-centered coordinates of latitude, longitude and elevation. Infospace identifies the location of a CSO’s service access ports, typically expressed in terms of Internet Protocol (IP), telecommunications and postal network addresses. Sociospace defines a CSO’s location within organizational (social) accountability structures [13] that encode formal or informal relations with other CSO, typically with respect to their authority over resources needed for value production.

CSO may be stationary or mobile in each dimension. If mobile, their coordinates will change with time. Resolving a CSO’s absolute and relative positions, velocities and accelerations requires a computable distance measure in each dimension. Given such a measure [4], and following standard notations in Newtonian mechanics, an incremental change in geospatial position is defined by the vector *dG*. Similarly, a change in infospatial location is given by *dI*, and a change in sociospatial position by *dS*. In keeping with standard physical principles, and accepting certain mathematical assumptions about differentiable (i.e., piecewise continuous) functions, a CSO’s velocity in each dimension is its rate of change in position over the period *dt*, or *dG/dt* (G´), *dI/dt* (I´) and *dS/dt* (S´), respectively. Similarly, its accelerations are the rates of change of velocity in each dimension, or dG^2/dt^2 (G´´), dI^2/dt^2 (I´´), and dS^2/dt^2 (S´´). Thus, in cyberspace a CSO in motion covers a distance *dC* over the period *dt* at a velocity of *dC/dt* (C´) and with an acceleration of dC^2/dt^2 (C´´).

While the notion of a physical object’s change in geospatial position is familiar, changes in its infospatial and sociospatial coordinates may be less intuitive. Infospatial motion is analogous to tracking mobile phones that drop then reestablishing their network addresses as they move among cell towers, possibly losing communications service for periods of time dependent on the phone’s geospatial velocity. In sociospace CSO roles and responsibilities will change as they switch from their roles in one federation context to another, perhaps as they join new supply chains or move from the role of consumer in one chain to that of producer, or go from superior in one context to subordinate in another. The velocity at which these context changes occur defines CSO agility and limits the number of federated alliances possible (i.e., its degree of multitasking).

For coherence and consistency (e.g., identity management, security, maintenance of historical records, etc.) CSO must maintain unique identities and occupy distinct locations in cyberspace. When CSO interact as trading partners [17] along a given supply or asset chain, they must be able to identify and validate each other’s credentials and relative locations. If a CSO is in motion, the nature of its interactions (e.g., quality of its services) may vary. For example, a change in location will typically affect logistics, such as the flow of physical or virtual resources (e.g., data) and affect their end-to-end propagation delays. A change in infospatial location will typically affect network routing and timing, while changes in a CSO’s sociospatial position may alter a federation’s command (authority) structure with corresponding changes in availability of resources. Changes in all three dimensions will affect and require attention to system security, consistency, integrity, exception handling and other critical governance considerations.

III. Forces acting in cyberspace

As suggested in Fig. 3, CSO exert *forces* on (i.e., affect behaviors of) other CSO through the sending and receiving of *messages*. As diagrammed, if message m_{ij}, issued at time t_k by CSO_i and directed at $CS0_j$, elicits a service response from CSO_j that is acknowledged in a return message m_{ji} issued at time t_k+dt, it will necessarily force (i.e., be accompanied by) allocation of some of CSO_j’s internal resources sufficient to meet the demand. In response, CSO_j’s acceptance and acknowledgement will alter the state of CSO_i. This is the cyberspatial analog of Newton’s third law of motion: a first body acting by a force on a second body will experience an equal and opposite reaction.

Note that in the interval *dt*, between message arrival and subsequent response, one or both interacting CSO may have altered their cyberspatial positions, being then separated by distance *dC*. As noted in Fig. 3, CSO_j moved a distance dC_{ij} relative to CSO_i along a trajectory v_j during period *dt*. This suggests that at the moment CSO issue messages they should include their cyberspatial position information to assist recipient CSO with authenticating and tracking coalition partners during periods of service.

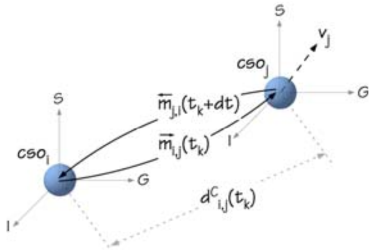

S
v_j
cso_j
G
S
$\overleftarrow{m}_{j,i}(t_k+dt)$
$\overrightarrow{m}_{i,j}(t_k)$
I
cso_i
G
$d^C_{i,j}(t_k)$
I

Figure 3 - Messages

Within federated systems messages are defined by strings encoded in an accepted alphabet and forming valid sentences defined by explicit rules of syntax (structure) and meaning semantics (meaning). Messages are issued to and received by each CSO through their published service access ports. Along supply chains messages carry requests from clients to service providers, with corresponding results subsequently returned by providers to their clients. Along the asset chains messages carry commands (e.g., policies, funding) issued by superiors to their subordinates and in return encode subordinates’ responses. These bilateral exchanges, governed by published trading protocols, establish the range of forces that may act on CSO.

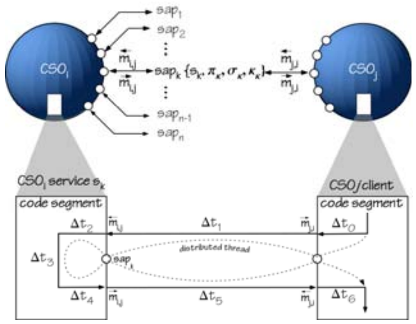

CSO_i
sap_1
sap_2
$\overleftarrow{m}_{i,j}$
$sap_k\ \{s_k, \pi_k, \sigma_k, \kappa_k\}$
$\overrightarrow{m}_{i,j}$
$\overleftarrow{m}_{j,i}$
$\overrightarrow{m}_{j,i}$
CSO_j
sap_{n-1}
sap_n
CSO_i service s_k
code segment
CSOj/client
code segment
Δt_2
$\overleftarrow{m}_{i,j}$
Δt_1
$\overleftarrow{m}_{j,i}$
Δt_0
Δt_3
sap_k
distributed thread
Δt_4
$\overrightarrow{m}_{i,j}$
Δt_5
$\overrightarrow{m}_{j,i}$
Δt_6

Figure 4 - Message Transport

End-to-end and round trip delays associated with CSO transactions (Fig. 4) are the sum of delays from each stage of a communications protocol. Round trip delays include time to encode and issue requests (Δt_0), transmission delay “on the wire” from sender to receiver (Δt_1), time to receive and decode request messages (Δt_2), time for the server to either reject the request or produce a result (Δt_3), time to encode and issue the result to the client (Δt_4), reverse transmission delay (Δt_5) and finally, time for the client to receive and absorb the result (Δt_6).

CSO may wish to block and await replies to their messages – a "synchronous" communications model. Alternatively, clients may wish to continue work on activities not dependent on the contents of a response, expecting to be signaled (i.e., interrupted) when subsequently results arrive – an "asynchronous" communications model. In the synchronous case, communication services support what is in effect a distributed process or *distributed thread* [2, 11] model, where the priority of a client's activity must be communicated to the service provider in order for it to support the client's completion-time requirements. This is analogous to standard notions of "just in time" production scheduling that seeks to remove as much "slack" as possible from horizontal or vertical value chains.

This phenomenon represents a challenging end-to-end timeliness (i.e., deadline scheduling) problem that occurs in distributed systems [11, 12, 15]. It arises when clients and servers along a supply chain each attempt to meet their individual completion-time requirements. In federated systems such self-optimization by participants typically results in sub-optimal behavior of the ensemble. Spontaneous arrival of messages introduces probabilistic demands on services resulting in resource scheduling conflicts compounding efforts to simultaneously optimize internal and group performance.

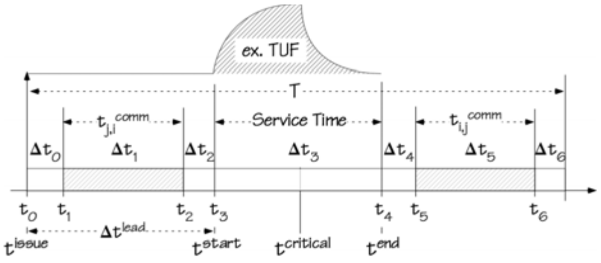
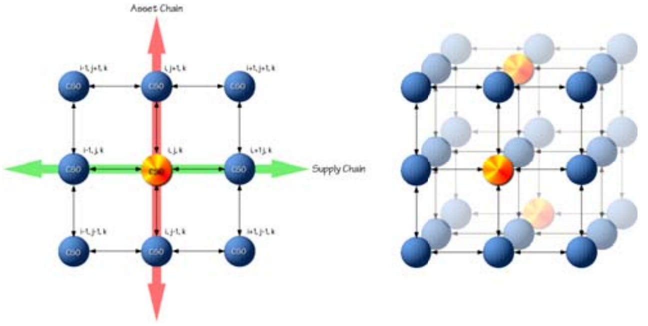


Figure 5 – Transaction Timing

With reference to Fig. 5, the critical step in maintaining end-to-end transaction timeliness (deadlines), and the only one reasonably predictable and solely within the purview of a service provider, is its service time (Δt_3). In the distributed thread model a client communicates its completion-time requirements as part of its service request. As shown, the CSO servicer can begin processing the service request as soon as it arrives (t_3), designated as t^{start}. In our model, CSO clients express their completion-time requirements in the form of *time-utility functions* (TUF) [14]. The serving CSO, in accepting a request, agrees make "best efforts" to complete service by no later than the deadline t_{end} while attempting to finish at the point where the TUF is maximized. In the example, the value of the service to the client begins at zero, rises to a peak, the time at which completing the service reaches its maximum value ($t^{critical}$), then falls exponentially to zero by t^{end}, the point at which the service no longer has value for the client. In effect, the mix of time-value functions at any given time establishes the CSO server's scheduling policy for maximizing the value for all competing client requests, and therefore its value to the federation.

For mobile CSO, the end-to-end transmission profile outlined above is affected in a number of quantifiable ways. First, transit times (Δt^{lead}) may vary widely depending on geospatial and infospatial distances between clients and servers. Second, infospatial access points may change and require (e.g., for security concerns) periodic revalidation. Third, sociospatial relations (e.g., accountability) may change or require that demand requests or service responses be authorized or have their priorities adjusted or deadlines delayed. While nominal end-to-end service guarantees may be covered by *service level agreements* between clients and servers, exigencies caused by cyberspatial motion may introduce unavoidable variations.



(A) **(B)**

Figure 6 - Federated Systems

In open ecosystems CSO coalesce to form communities of mutual interest supporting their respective value propositions. Some are strategic, renewable and long-lived, while others are spontaneous, tactical and of shorter duration. These communities are often referred to as federations since they comprise groups of sovereign entities with mutually agreed upon and codified rules of engagement (e.g., contracts). Such ecosystems naturally form along gradients of increasing value, vertically along asset chains and horizontally along supply chains. In this view value chains resemble lattice structures (Fig. 6A). Further, CSO may simultaneously hold membership in more than one federation (Fig. 6B) at a given time, and perform different roles in each. In these situations (e.g., participation in multiple supply chains) effective governance requires an agile and reliable mechanism for a CSO to context-switch among federations and roles while simultaneously meeting completion-time commitments in each. This multitasking requirement is a major source of "faults" in poorly designed governance systems. It is completely analogous to the basic role of operating systems in networks of multiprogrammed computer systems.

IV. FEDERATION STRUCTURE

In addition to a CSO's external associations (Fig. 6), it may contain an arbitrarily complex internal management accountability structure (Fig. 7) [13]. This additional dimension represents a CSO's sociospatial relations existing among its internal functional elements, some of which may be cost centers and some profit centers. Internal profit centers, by reference to their own value propositions are by definition encapsulated (“nested”) CSO.

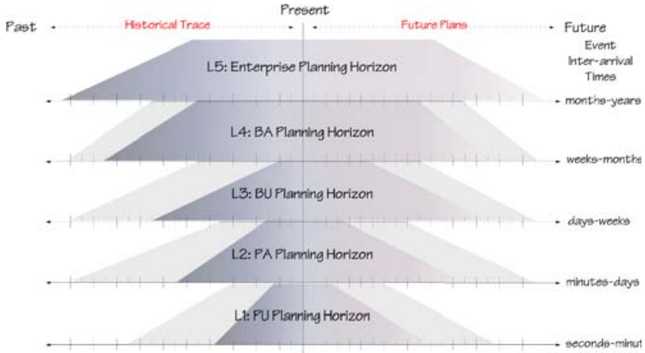


Figure 7 – Planning Horizons

Fig. 7 and 8 diagram typical enterprise governance (i.e., sociospatial) levels and their respective planning horizons, with history to the left and future to the right, including their respective timing considerations. Fig. 7 identifies business areas (BA), each typically containing multiple business units (BU). A BU may contain several production areas (PA), or *factories*, each in turn containing one or more production units (PU), and so forth. In usual accountability hierarchies, each level is responsible for self-regulation (e.g., managing its personnel, facilities, inventories and machines). Each is also accountable to policies issued at levels above and in turn responsible for policies governing behaviors at levels below.

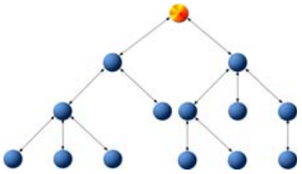


Figure 8 - CSO Structure

Value propositions govern the behaviors of CSO operating at the various levels of a federation, each responsible for serving clients with competitive services. As noted, the lower the level a CSO functions the more specialized are its services, the smaller its data sets and the faster it responds. While the horizontal performance of lower-level supply chains is generally understood, performance of the vertical asset chain defining an enterprise’s sociospatial structure may be less so. This is often due to the lack of operational formality in human-human (i.e., socio-political) interactions. Including this dimension in our governance structure represents an important contribution to the theory of federated governance systems.

V. The Governance Challenge

To appreciate the distributed governance challenge we must look more closely at the inner workings of a cyberspatial object. As noted previously, CSO governance involves two critical subordinate tasks (modeled in Fig. 9), one managing transactions along its supply chain and the other managing transactions along its asset chain. Coordination of these two internal processes is the essence of management for sustainable value production. These are the essential elements for achieving dynamic stability, the locus of effective governance.

This recursive model of enterprise governance recognizes a CSO’s internal structure. As is typical of governmental, educational, military and commercial organizations it recognizes and, according to principles of Management Cybernetics [6, 10], formalizes the roles and responsibilities of three primary sociospatial actors, labeled here by their levels (echelons) of responsibility: E5, E4 and E3. E5 (director, commander, CEO) represents the highest level of authority within a CSO. E4 (navigator, Chief of Staff, CTO) provides strategy, analysis and planning. And E3 (operator, XO, COO) attends to detailed operational matters. Important but ancillary roles reporting to E3 include E2 (coordinators, schedulers, project managers) dealing with synchronization of concurrent activities that compete for shared resources, E1 (controllers, managers, supervisors) responsible for the execution of specific tasks, E0 (tasks, processes) defining the value production processes under E1’s direct control, and E3* (auditor, monitor, observer) responsible for independent assessment and reporting of value production activities. Note that within this governance structure there are two complementary control loops: one *regulatory* in nature expressed by the E0-E1-E2 loop and one *supervisory* expressed in the E3-E2-E1 loop. These two complementary feedback loops (Figure 10) control E3 and E1 operations that are qualified by policies (i.e., strategies) issued by the combined E5-E4-E3 executive function. The natural tensions between the two loops provide the balance needed efficiently allocate energy and resources needed for dynamic stability (homeostasis).

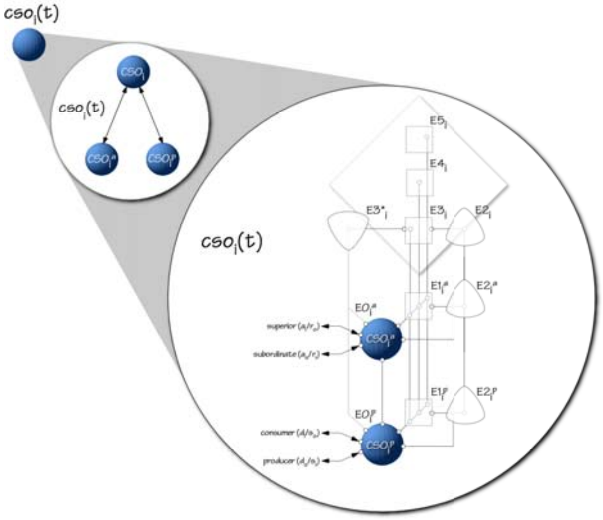


Figure 9 - CSO Governance Structure

Regulatory and supervisory controls are specialized applications of a more general cyclic or “canonical” feedback mechanism whose basic operation is shown in Fig. 10. It includes functions for i) measuring the activities of some real or synthetic process of interest in order to

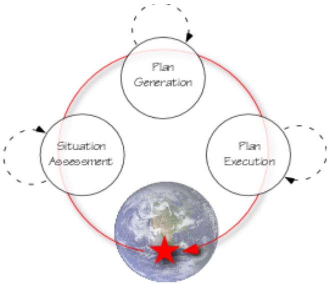


Figure 10 – Canonical Control

determine its status, ii) generating one or more feasible plans for responding to situations of concern, iii) selecting, assigning resources to and executing the plans of action, and iv) cycling back to monitor the effects of such interventions. Regulatory and supervisory control strategies may vary widely depending on value proposition and context, application level, complexity, safety, availability requirements of processes under control, fidelity, accuracy of available measurements and the speed and precision required of effective and stable control.

Placing the canonical control loop into our CSO model yields the structure in Fig. 11, here with the loop represented by a dashed line responsible for value production along the CSO_i^p (aka, $E0_i^p$) supply chain. This construction highlights a key unit of governance within a CSO. This process registers consumer demand, subsequently issues demand orders on its suppliers, receives those supplies and subsequently produces results for its customers. Management of value production is provided by the E0-E1-E2 loop regulating consumption of internal resources as it responds the demands along the supply chain. The goal of the loop is to minimize variety (error) in service through optimal resource scheduling.

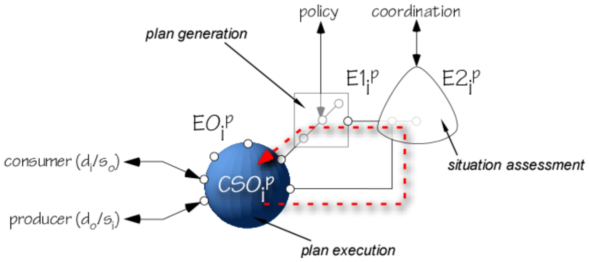


Figure 11 – Supply Chain Controller

Synchronized supply and asset chain control is diagrammed in Fig. 12. Assets flowing along CSO asset chains are required to meet time-varying obligations for services delivered along supply chains. Finance, as taught in business schools, typically accounts for these flows on balance sheets and income statements, respectively. Coordination of these simultaneous and competing activities is the essential role of supervisory control, the purview of executive management in balancing (optimizing) activities along both axes.

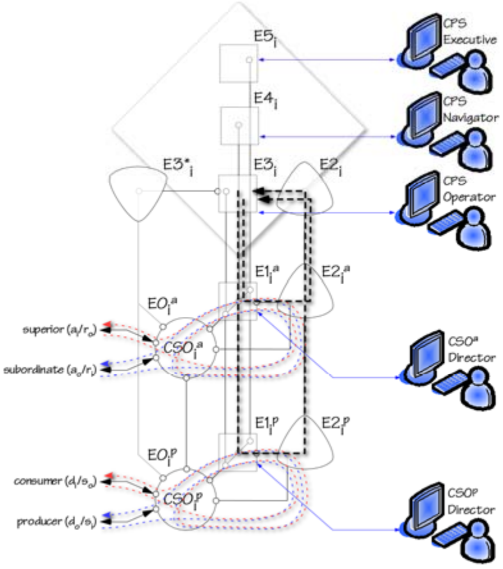


Figure 12 – CSO Coordination & Control

In Fig. 12 horizontal production loops are coordinated through vertical supervisory loops expressed as dashed lines coupling E1, E2 and E3. Among other duties, supervision seeks to rationalize the scheduling (timing and allocation) of shared resources and establishment of priorities. The situation is more complex when a given CSO serves multiple supply chains or participates in multiple federations.

A key characteristic of our model, as expressed by the symbols used in its expression, lies in its ability to scale – its recursive nature. As noted, control of value production takes place along supply and asset chains among CSO operating at various levels in an enterprise, from its lowest levels to its highest levels of production. To be generally applicable, the model must apply equally well to all levels of concern. The symbols used in these diagrams express this recursive nature of control as follows.

The rectangle used to represent E1, the regulatory controller for the E0 unit of value production, is intentionally similar in shape to the rectangle encapsulating the E5-E4-E3 management triumvirate, albeit pivoted 45 degrees. This construction is meant to emphasize the nested (nested, recursive, fractal) nature of governance. At a given level of an enterprise (e.g., L3 in Fig. 7) the E1 regulatory controller is equivalent to E5-E4-E3 supervisory controller at the next lower level (e.g., L2). This recursive structure defines a key design pattern of our scalable governance framework, where management services remain applicable (i.e., are reusable) throughout. Further, this construction encourages incremental implementations, permitting automation of one CSO at one level of concern to be followed by subsequent implementations in other CSO elsewhere along supply or asset chains. Fig. 13 makes explicit this nested geometric structure, showing a parent CSO in its relation to subordinate CSO along its asset chains.

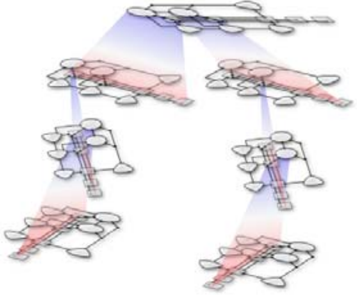


Figure 13 – Embedded Systems

We end this introduction on our governance framework for distributed systems with a few remarks on interfaces to human operators. In effect, each point of control, from E5 to E0 (Fig. 12), supports user interfaces (UI) to their relevant monitoring and control functions. UI provide access to performance indices and allow adjustments to policies and operating set-points. Hence, UI are critical, if for no other reason than they support through standards fidelity and transparency of operations that permit personnel to be deployed, effectively function and interact within and across multiple levels of an organization [5].

By analogy, the Federal Aviation Administration enforces airplane cockpit standards allowing pilots to fly and navigate even when operating a wide range of aircraft types. Similarly, the National Highway Traffic Safety Administration enforces automobile cockpit standards allowing people to drive a variety of car models. Such standards permit a high degree of innovation in such physical CSO as airplanes, automobiles, manufacturing plants, entertainment electronics, and computers while enforcing operational consistency and safety in their use.

VI. CONCLUSION

This paper introduced a scientific and engineering framework for implementing distributed time-critical service systems [18], systems whose behaviors unfold in cyberspace. This is the third in a sequence of formal presentations [4, 5] of a design framework, extending those previous attempts while explicitly dealing with automation and control of intelligent objects whose operating context has geospatial, infospatial and sociospatial extents. Further, it refines our earlier theory of cyberspatial mechanics, including semantics that formally deal with the dynamics of information systems expressed, at least in part, by their relative positions and motions – characteristics useful in specifying and enforcing such distributed system requirements as security, identity, and logistics governing information assets, their availability, the meaning of faults and their effective management.

REFERENCES

1. J. Albus, "Outline for a Theory of Intelligence," IEEE Trans System, Man and Cybernetics, Vol. 21, No. 3, June 1991
2. J.S. Anderson and E.D. Jensen, "Distributed Real-Time Specification of Java (DRTSJ)—A Status Report (Digest)," *JTRES'06*, October 11-13, 2006 Paris, France
3. R. Ashby, *Introduction to Cybernetics*, Chapman Hall, 1957
4. J.S. Bayne, "Cyberspatial Mechanics," IEEE Transactions on Systems, Man and Cybernetics – Part B, Vol. 38, No. 3, June 2008
5. J.S. Bayne, *Creating Rational Organizations—Theory of Enterprise Command and Control*, Café Press, September 2006
6. S. Beer, *The Brain of the Firm*, Wiley, 1994
7. S. Beer, *Decision and Control*, Wiley, 1988
8. R.C. Conant, "Laws of Information Which Govern Systems," IEEE Trans of Systems, Man and Cybernetics, Vol. 6, No. 4, 1976
9. T. Erl, *Service-Oriented Architectur*e, Prentice-Hall, 2005
10. J. Forrester, *Collected Papers*, Pegasus Communications, 1975 and http://www.systemdynamics.org/
11. http://www.real-time.org
12. http://www.rtsj.org/
13. E. Jaques, *Requisite Organization*, Cason Hall, 1992
14. E.E. Jensen, "Utility Functions: A General Scalable Technology for Software Execution Timeliness as a Quality of Service," *Proc. Software Technology Conf.*, Utah State Univ., April 2000
15. P. Li, "Utility Accrual Real-Time Scheduling: Models and Algorithms," *PhD Thesis*, Virginia Polytechnic & State University, 2004
16. K. Merchant and W. Van der Stede, *Management Control Systems*, Prentice Hall, 2003
17. Object Management Group (OMG), "Real-Time CORBA Specification," V1.2, http://www.omg.org/cgi-bin/doc?formal/05-01-04
18. J. Spohrer and D. Riecken, "Special Issue: Services Science," Communications of the ACM, July 2006
19. L. Whitman and B. Huff, "The Living Enterprise Model," Automation and Robotics Research Institute, U Texas at Arlington, 2000
20. N. Wiener, *Cybernetics*, MIT Press, 1948
21. M. Wooldridge, *Reasoning About Rational Agents*, MIT Press, 2000

Jay S. Bayne (SM'96) received a B.Sc. degree in electrical engineering in 1970, a M.Sc. degree in electrical engineering and computer science in 1971, and a Ph.D. degree in electrical engineering and computer science in 1976, all from the University of California at Santa Barbara. He was Professor of Computer Science at California Polytechnic State University at San Luis Obispo from 1973-1984. It was there he founded his first company, Protocol Solutions, Inc. He has since held VP Technology and Strategic Marketing positions for ABB and JCI, global engineering companies involved in the design and application of distributed, fault-tolerant, real-time control systems for automation of transportation, power production and distribution, pulp and paper, chemical, pharmaceutical, petroleum, building environments and related processes. He is presently Adjunct Professor of Computer Science at the University of Wisconsin, a consultant to the Office of the Assistance Secretary of Defense (OSD/NII) and Chief Executive of his second company, Meta Command Systems, Inc. He is Chairman and Executive Director of the Milwaukee Institute, a nonprofit computational research and technology institute. Dr. Bayne has published numerous papers and written and contributed to several books on the subject of the theory and technology of real-time high-availability systems.