Architecting Principles for Systems-of-Systems

Mark W. Maier

CH1-460, *The Aerospace Corporation, Chantilly,* VA 20151

Received April 20, 1997; Revised August 4, 1998; Accepted November 3, 1998

ABSTRACT

While the phrase "system-of-systems" is commonly seen, there is less agreement on what they are, how they may be distinguished from "conventional" systems, or how their development differs from other systems. This paper proposes a definition, a limited taxonomy, and a basic set of architecting principles to assist in their design. As it turns out, the term system-of-systems is infelicitous for the taxonomic grouping. The grouping might be better termed "collaborative systems." The paper also discusses the value of recognizing the classification in system design, and some of the problems induced by misclassification. One consequence of the classification is the identification of principal structuring heuristics for system-of-systems. Another is an understanding that, in most cases, the architecture of a system-of-systems is communications. The architecture is nonphysical, it is the set of standards that allow meaningful communication among the components. This is illustrated through existing and proposed systems. © 1999 John Wiley & Sons, Inc. Syst Eng 1: 267–284, 1998

1. SUMMARY

While the term "system-of-systems" has no widely accepted definition, the notion is widespread and generally recognized. There is an emergent class of systems that are built from components which are large-scale systems in their own right. Prominent existing examples include integrated air defense networks, the Internet, and enterprise information networks. Much larger, more complex and distributed examples are being planned. Notable among them are intelligent transport systems [IVHS America, 1992; USDOT, 1995], military C4I and Integrated Battlespace [Butler et al. 1996], global satellite networks [Stuart, 1996], and partially autonomous flexible manufacturing systems [Hayes, 1988].[1]

The question addressed by this paper is whether or not there is a useful taxonomic distinction between various complex, large-scale systems that are commonly referred to as "systems-of-systems." For there to be a useful taxonomic distinction, we should be able to divide systems of interest into two (or more) classes such that the members of each class share distinct attributes, and whose design, development, or operations pose distinct demands. This paper argues that there is a useful taxonomic distinction, and that the distinction is based on the operational and managerial independence of the systems components. By the tenets of this paper, a system would be termed a "system-of-

[1] An earlier version of this paper appeared in the Proceedings of the 1996 Symposium of the International Council on Systems Engineering.

systems" or a "collaborative system" when: (1) Its components) fulfilled valid purposes in their own right and continued to operate to fulfill those purposes if disassembled from the overall system, and (2) the components systems are managed (at least in part) for their own purposes rather than the purposes of the whole. Moreover, this paper argues that the commonly cited characteristics of systems-of-systems (complexity of the component systems and geographic distribution) are not the appropriate taxonomic classifiers. The principal reason is that there are design guidelines that address those demands that apply differently for systems within and without the proposed class.

Within these properties are further taxonomic divisions, although they carry less importance in development. For example, there is a distinction between collaborative systems that are organized and managed to express particular functions or purposes, and those in which desired behaviors must emerge through voluntary and collaborative interaction without central direction.

Many problems in developing systems-of-systems can be traced to misclassification, either as to monolithic system versus system-of-systems or as to category within system-of-systems. The misclassification issues are related to enabling and fostering collaboration, not to provisions for geographic distribution or complexity issues. Especially important is a failure to architect for robust collaboration when direct control is impossible. This arises when the developers believe they have greater control over the evolution of the system-of-systems than they actually do. In believing this they may fail to ensure that critical properties or elements will be incorporated by failing to provide a mechanism matched to the problem.

The independence and extent of these aggregations of systems results in an even greater emphasis on interface design than in traditional system architecting and engineering. Since the components are often developed independently of the aggregate, the aggregate emerges as a system in its own right only through the interaction of the components. Because elements will be independently developed and operated, the system-of-systems architect must express an overall structure largely (or even wholly) through the specification of communication standards.

Systems-of-systems are largely defined by interface standards. Generally, these will be communication protocol standards, possibly at many levels of a layered communication model. Only in special cases will there be more concrete interfaces. Different problems require interface standards at different levels. Some applications, an intelligent transport example stands out, can require a unique standard built from physical transmission upward. As data communication becomes ubiquitous, however, the standards that enable each particular system-of-systems will be high-level standards, operating above the transport layer, which define the semantic content of messages passed among the components. These standards, sometimes referred to as middleware, will most likely be built on distributed object and messaging frameworks. Some cases, military systems for example, may define part of their architecture through human training and indoctrination.

2. DISCUSSION

This analysis of systems-of-systems architecting divides into eight parts. The first part reviews some literature discussions of "system-of-systems" and introduces two commonly cited examples, integrated air defense systems and the Internet. Second, the paper provides a two-part core definition for "system-of-systems" viewed as a taxonomic node. Third, the paper examines several design heuristics that are of special importance in systems meeting the definition. Fourth, an argument is framed for the taxonomic node and the examples are reexamined in light of the heuristics in the fifth part. This part also introduces a future example of a system-of-systems, the Intelligent Transport Systems. Sixth, we examine some additional taxonomic division. In the seventh part we examine the impact of misclassification (which furthers the overall argument). Lastly, the design heuristics and past experience are combined to consider communications as the architecture[2] of a system-of-systems.

2.1. What Is a "System of Systems?"

While the term "System of Systems" appears frequently, there is no widely accepted definition of its meaning. The use of a separate term "system-of-systems" implies a taxonomic grouping. It implies the existence of distinct classes within systems. Such classes are useful for engineering only if they represent distinct demands in design, development, or operation. In a formal sense, system-of-systems is not descriptive. A system is generally understood to be an assemblage of components that produces behavior or function not available from any component individually. The IEEE 610.12 definition is representative. The INCOSE definition is similar:

[2]Throughout this paper, "architecture" is used in the sense of a fundamental or unifying system structure defined in any system dimension or view. This is a sense of architecture reflected in the INCOSE System Architecture Working Group definition as of 1998, the sense of the IEEE Architecture Working Group definition current as of 1998, and the author's previous work.

> System: a collection of components organized to accomplish a specific function or set of functions (IEEE)

Under this definition, a personal computer is a system. The computer's disk drive, video monitor, processor, and so forth are likewise systems. So also, formally, a personal computer is a "system-of-systems" because it is an assemblage of components that are individually regarded as systems. Thus, formally, the term system-of-systems has no distinguishing power. Its broad use, however, is suggestive that investigators have found value in distinguishing very large and distributed systems from much less complex and more compact systems. "System-of-systems," as commonly used, suggests assemblages of components that are themselves significantly complex, enough so that they may be regarded as systems and that are assembled into a larger system. Many authors, however, prefer the notion of geographic distribution to a notion of a type of interrelationship.

At least two previous authors have proposed taxonomies that carry the notion of a distinct class of systems generally characterized as large, complex, geographically distributed, and composed of components that are significant systems in their own right. Shenhar [1994, p. 268] proposed a two-dimensional system taxonomy that includes a category called an "array." An array system in Shenhar's terms is:

> A large widespread collection or network of systems functioning together to achieve a common purpose.

Similarly, Eisner [1993] defines systems-of-systems as large geographically distributed assemblages, but envisions only centrally directed development efforts in which the component systems and their integration are deliberately, and centrally, planned for a particular purpose. Thus both Shenhar and Eisner use the term "system-of-systems" to describe geographically distributed systems which are otherwise developed and managed conventionally. Shenhar does make distinctions in best practices for development for what he calls an array that differ from best practices for nondistributed systems. However, in all of these cases the development and operations management model is fundamentally centralized.

Shenhar comes closest to making a case for a useful taxonomic node by discovering best practices different from those associated with his other taxonomic nodes. However, neither geographic distribution nor the complexity of components meets the test of being discriminating characteristics for distinctly different design approaches, when the appropriate examples are considered.

A number of important existing and emerging systems are not characterized by central management of either development or operations. These systems are commonly, though not always, geographically distributed and evolutionary. What is unique about them is their fundamentally collaborative rather than directed structure. These systems are composed of subsystems that are capable of operating independently of the integrated whole, and do operate in partial independence as part of normal operations. The integrated system exists because of deliberate decisions by the subsystem or component developers to collaborate as part of a greater whole, and that decision to collaborate is an ongoing one. Another term sometimes used for collaborative assemblages is "federated system." The next sections consider wide area communication networks and integrated air defense systems.

Wide Area Networks

All wide area network systems are geographically distributed and have complex components. Hence, they fit the common usage of system-of-systems. Major examples include IBM System Network Architecture (SNA), the Bell telephone system, Asynchronous Transfer Mode networks, and the Internet. However, these examples are distinguished by the degree of centralization of control in their design and operation. The Internet is the classic example of distributed control and operation. The central architecture of the Internet, in the sense of an organizing or cohering structure, is a set of protocols now called TCP/IP. Their relationship to other protocols commonly encountered in the Internet is shown in , modeled after [Peterson and Davie, 1996: 37]. The TCP/IP suite includes the IP, TCP, and UDP protocols in Figure 1.

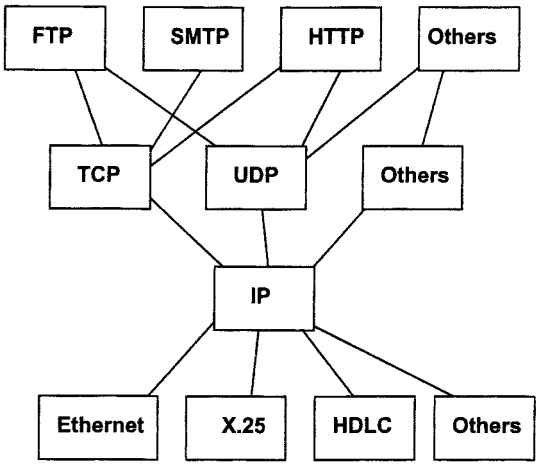

Figure 1 Internet protocol dependencies.

The TCP/IP family protocols are based on distributed operation and management. All data are encapsulated in packets, which are independently forwarded through the Internet. Routing decisions are made locally at each routing node. Each routing node develops its own estimate of the connection state of the system through the exchange of routing messages (also encapsulated at IP packets). The distributed estimate of connection state are not, and need not be consistent or complete. Forwarding is designed to be acceptably successful even when connection state estimates are inconsistent and incomplete.

The distributed nature of routing information, and the memoryless forwarding, allows the Internet to operate without central control or direction. A decentralized development community matches this decentralized architecture. There is no central body with coercive power to issue or enforce standards. The development community creates standards through collaborative arrangements, the Internet Engineering Task Force (IETF) and related bodies, and deploys them in a similar fashion.

The telephone system, and the SNA computer networking system, in contrast, rely on centralized control over operation and development. They can also more efficiently utilize transmission technologies or more efficiently carry out specific applications since they can tie directly to them. Nodes in the system are centrally directed in maintaining consistent and complete connection state models. Instead of memoryless packet forwarding, these networks use end-to-end connection abstractions. They also tie more closely to both the transmission technology and user application, and derive efficiency from doing so.

Comparison of the three example systems (and a further comparison to ATM) reveals the important interplay between distributed operation and development. The Internet can be developed in a collaborative way largely *because* its operation is collaborative. Because the Internet uses best-effort forwarding and distributed routing, it can easily offer new services without changing the underlying protocols. Those new services can be implemented and deployed by groups that have no involvement in developing or operating the underlying protocols; but only so long as those new services do not require any new underlying services. So, for example, groups were able to develop and deploy IP-Phone (a voice over the Internet application) without any cooperation from TCP/IP developers or even Internet service providers. However, the IP-Phone application cannot offer any quality of service guarantees since the protocols it is built on do not offer simultaneous delay and error rate bounding.

In contrast, the centralized protocols like the telephone system and SNA can offer richer building block network services, especially including quality of service guarantees. However, they are much less able to allow distributed operation, and cannot permit distributed development. Strict adherence to protocols throughout the layer stack is required. Since telephone network equipment is developed by many firms, the industry has produced much stronger standards organizations, with much stronger centralized control. Centralized computer network standards, like SNA, have largely fallen into disfavor because they could not easily adapt to the kinds of service desired by computer users, and their centralized architectures could not rapidly adapt. The telephone system has been, and continues to be, highly successful. However, its basic architecture of switching and close ties to transmission technology is under strong challenge.

The two system types differ in their robustness against accidental and deliberate attack. In principle, a decentralized system like the Internet should be less vulnerable to destructive collective phenomena and be able to locally adapt around problems. In practice, both the Internet with its distributed control model and the telephone system with its greater centralization have proven vulnerable to collective phenomena. It turns out that distributed control protocols like TCP/IP are very prone to collective phenomena in both transmission and routing [Bertsekas and Gallager, 1992: Chap. 6]. Careful design and selection of parameters has been necessary to avoid network collapse phenomena. One reason is that the Internet uses a "good intentions" model for distributed control which is vulnerable to nodes that misbehave either accidentally or deliberately. There are algorithms known which are robust against bad intentions faults, but they have not been incorporated into network designs.

Wide area telephone blackouts have attracted media attention and shown that the more centralized model is also vulnerable. The argument about decentralized versus centralized fault tolerance has a long history in the electric power industry, and even today has not reached full resolution.

The ATM standard tries to combine characteristics of both centralized and decentralized architecture. It adopts packet forwarding as the transfer model, but uses virtual circuit switching for the forwarding rule, which requires networkwide consistency on all connections. The ambition for ATM has been to combine the best features of both the computer and telephone networking worlds and offer a rich set of services including both guaranteed and best effort services. The protocol standard has also been developed in an international collabo-

rative model without a central entity exercising control over decisions.

While the situation is still a work in progress, actual ATM products do not appear to have effectively combined the two models. Most ATM deployments are basically centralized and operate as a flexible means of building point-to-point high-speed connections. The service abstractions planned to combine packetlike flexibility in resources on-demand with service guarantees have not been implemented, apparently because they do not work. The difficulty in combining the centrally controlled and highly distributed models is further evidence for the linkage of the two proposed criteria. Managerial and operational independence work together, and represent an alternative approach to structuring systems.

Integrated Air Defense

An integrated air defense network is an assemblage of radars and other sensors, missiles and other weapons, command nodes, and communication networks tying all the components together. An integrated air defense system need not be geographically distributed (think of a complex warship). An integrated air defense system can utilize varying degrees of centralization in control. The components can be closely tied together, in which case all sensor data would be fused into a single battlespace picture, and all weapons would be centrally commanded in response to that battlespace picture. In contrast, the system could be highly distributed with each weapon having organic sensor capabilities and operating independently, constrained only by predetermined rules-of-engagement. Intermediate levels of control centralization can readily be imagined. Likewise, there are varying degrees of centralization possible in development and management. All elements could be under a central acquisition authority with full budget and engineering authority, or each element could be developed by an independent entity with its own financial and technical resources, or there are countless intermediate possibilities.

Operationally, the two extremes clearly have different properties. The centralized system can behave in ways impossible for the completely decentralized system, for example, cross-cueing weapons and optimizing weapon selections from the whole battle situation. On the other hand, the centralized system is vulnerable to information saturation, or disruption by attack on the command centers in ways that the decentralized system is not vulnerable to. Likewise, centralized versus decentralized development will show opportunities for systemwide optimization and vulnerability to single points of failure. In the discussion to follow, the centralization of control and management will be studied as the central discriminant of classes of system-of-system.

2.2. Collaborative System-of-Systems Definition

It is collaboratively integrated systems for which this paper proposes the term "system-of-systems." This paper proposes two principal distinguishing characteristics for applying the term "systems-of-systems," or alternatively the term "collaborative system." A system that passes these two criteria is designated a "system-of-systems." A system that does not meet these two criteria is not considered a system-of-systems under this definition, *regardless* of the complexity or geographic distribution of its components.

A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possesses two additional properties:

Operational Independence of the Components: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.

Managerial Independence of the Components: The component systems not only *can* operate independently, they *do* operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.

By the criteria of this paper, a system that has operational and managerial independence of its elements is a system-of-systems. But a system composed of complex subsystems that do not have both operational and managerial independence is not a "system-of-systems," no matter the complexity of the subsystems. So, for example, an integrated air defense network without a single acquisition or command authority is a system-of-systems, while an individual missile system is not. The air defense network would not be a system-of-systems if acquired and run by a single entity that carefully expunges any redundancy that would allow its elements to be run independently of the integrated whole. The integrated air defense network would be termed a system-of-systems here if its components (radars, missile batteries, command centers, etc.) can and do operate independently. An individual missile system is not termed a "system-of-systems" in this paper because its components (motor, body, sensor, etc.) do not possess both operational and managerial independence.

One observation about systems meeting these criteria can be made immediately: They are always more costly than a system not meeting these criteria designed to fulfill the integrated system's purpose. If a system-of-systems fulfills a purpose P, it will also fulfill the additional purposes of the individual elements. A "monolithic system" (one not meeting criteria 1 and 2 above) could thus be designed to full purpose P and not any of the component system purposes at presumptively less cost. The higher costs of a system-of-systems are because of their inherent redundancy. Since the components can operate independently they possess capabilities duplicated in other components. By eliminating that redundancy one could reduce costs.

As a result, a system-of-systems may arise partially by accident (in which case no cost minimization criteria would been applied) or deliberately in cases where the side benefits of maintaining the redundancy outweigh any desire to minimize cost. One reason for this is when the disaggregated operational modes carry value themselves that outweigh the additional cost. Another reason is when the total system-of-systems cost is not borne by a single identifiable customer and so there is no decision-maker to whom minimizing total cost is important.

The basic argument for these criteria as defining a taxonomic node is twofold. First, we observe that there are architectural principles widely employed in successful systems meeting these criteria that are not nearly as important or take on different forms in systems not meeting these criteria. Second, systems that are misclassified by meeting only one of the two criteria are typically troubled, and are troubled in distinctive patterns.

2.3. Architectural Principles

Having a proposed definition for systems-of-systems, we turn to observations of design principles. Design principles, heuristics, best practices, and patterns are all similar terms for the idea that soft rules correlated with success can be inducted from observing system development. All the design principles considered here were originally published as heuristics in Rechtin [1991] and refined in Maier [1994] and Rechtin [1997]. They are further refined here as appropriate.

Stable Intermediate Forms

The heuristic on stable intermediate forms is given in Rechtin [1991, p. 91] as:

> Complex systems will develop and evolve within an overall architecture much more rapidly if there are stable intermediate forms than if there are not.

The origin of this heuristic is civil construction where it has been recognized as desirable for a building under construction to be self-supporting at many stages in its erection. Systems meeting the proposed system-of-systems criteria are fundamentally collaborative. It cannot be assured that all participants will continuously collaborate, and evolution based on new self-assessments of their objectives for collaboration should be assumed. Even systems-of-systems with considerable central direction, like an integrated air defense system, may be exposed to sudden (and violent) "reconfiguration." The examples to follow all exhibit a broad set of stable intermediate forms, both in time and in spatial deployment.

Taken more generally on systems, stability means intermediate systems should be capable of operating and fulfilling useful purposes before full deployment or construction is achieved. A more general interpretation is that intermediate forms should be technically, economically, and politically self-supporting. It should be possible to build and operate the intermediate forms within the economic and political framework of the planned full system.

Technical stability means that the system operates to fulfill useful purposes. Economic stability means that the system generates and captures revenue streams adequate to maintain its operation. Moreover, it should be in the economic interests of each participant to continue to operate rather than disengage. Political stability can be stated as the system has a politically decisive constituency supporting its continued operation [Rechtin, 1997: Chap. 10].

All of the examples show this heuristic at work. Integrated air defense systems are designed with numerous fall back modes, down to the antiaircraft gunner working on his own with a pair of binoculars. The Internet allows components nodes to attach and detach at will. A still existing subset of the net [the UNIX-to-UNIX Copy Protocol (UUCP) system] is based on intermittent telephone modem connections among its members. The ITS (an example to come) will be deployed piecemeal and unevenly based on the preferences of local and state governments and the willingness of the public to invest in in-car systems. At least in the United States, a monolithic ITS with a distinct startup date is impractical.

A corollary is that components should be severable from the system-of-systems without destroying the desired emergent behaviors. Since the components are at least partially independent in their operation and development, there can be no guarantee of their availability.

Policy Triage

This heuristic gives guidance in selecting and supporting components for a system-of-systems. It is given in [Rechtin, 1991, p. 83] as:

> The triage: Let the dying die. Ignore those who will recover on their own. And treat only those who would die without help.

The central distinction between systems meeting the proposed criteria and those that do not is the scope of control of the development team. On a system-of-systems the development team does not fully control either the development or the modes of operation of the target system. A system-of-systems design team must employ triage, where a system team (having full control over the components under his paper definition) does not have to. The need for triage is a distinguishing characteristic. The design guidance is to choose very carefully what to try and control. Attempting to overcontrol will fail for lack of authority. Undercontrol will eliminate the system nature of the integrated system.

Classic examples of good triage choice are in technical standards. For example, the Motion Picture Experts Group (MPEG) chose to only standardize the information needed to decompress a digital video stream [Chiariglione, 1998]. The standard defines the format of the data stream, and the operations required to reconstruct the stream of moving picture frames. However, the compression process is deliberately left undefined. By standardizing decompression the usefulness of the standard for interoperability was assured. By not standardizing compression the standard leaves open a broad area for the firms collaborating on the standard to continue to compete. Interoperability increases the size of the market, a benefit to the whole collaborative group, while retaining a space for competition eliminates a reason to not collaborate with the group. Broad collaboration was essential both to ensure a large market, and to ensure that the requisite intellectual property would be offered for license by the participants.

Leverage at the Interfaces

Two heuristics, here combined, discuss the power of the interfaces:

> The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces.

When the components of a system-of-systems are highly independent, operationally and managerially, the architecture of the system-of-systems *is* the interfaces. There is nothing else to architect. The Internet *is* the interfaces, in this case the Internet Protocol (IP). An integrated air defense system, in the sense of an system above the independent elements, *is* the command, control, and communications network.

Thus, the design history of successful systems-of-system should show much higher attention to the interfaces than to the components. This is certainly true in the case of the Internet. The Internet oversight bodies concern themselves almost exclusively with interface standards. Neither physical interconnections nor applications above the network protocol layers are standardized. This leads to interesting distinctions between design practices for conventional systems and systems-of-systems (taking the two criteria here as defining). In a system-of-systems issues like lifecycle cost are of very low importance. The components are developed collaboratively by the participants, who make choices to do so independently of any central oversight body. The central design team cannot choose to minimize life cycle cost, nor should they, because the decisions that determine costs are outside their scope. The central design team can choose interface standards, and can choose them to maximize the opportunities for participants to find individually beneficial investment strategies.

Ensuring Cooperation

> If a system requires voluntary collaboration, the mechanism and incentives for that collaboration must be designed in.

In a system-of-systems the components, at least to a degree, actively choose to participate or not. Like a market, the resulting system is the web of individual decisions by the participants. Thus, the economists' argument that the costs and benefits of collaboration should be superior to the costs and benefits of independence for each participant individually should apply. As an example, the Internet maintains this condition because the cost of collaboration is relatively low (using compliant equipment and following addressing rules) and the benefits are high (access to the backbone networks). Similarly in MPEG video standards, compliance costs can be made low if intellectual property is pooled, and the benefits are high if the targeted market is larger than the participants could achieve with proprietary products. Without the ability to retain a competitive space in the market (through differentiation on compression in the case of MPEG [Chiariglione, 1998]), the balance might have been different. Alternatively, the cost of noncompliance can be made high, though this method is less used.

An alternative means of ensuring collaboration is to produce a situation in which each participant's well being is partially dependent on the well-being on the other participants. This joint utility approach is known, theoretically, to produce consistent behavior in groups [Brock and Durlauf, 1995]. A number of social mechanisms can be thought of as using this principal. For example, strong social indoctrination in military training ties the individual to the group and serves as a coordinating operational mechanism in integrated air defense.

2.4. Argument for the Taxonomic Node

If systems that possess the two properties asserted above do form a valid taxonomic node, then we should see distinctly different choices in design, development, and operations in similar alternative systems, one of which meets the criteria and one of which does not. If the two criteria above are "better" taxonomic criteria than the informal criteria in the literature (geographic distribution and complexity of the elements), then the informal criteria should fail to substantially distinguish differences in system structure or development.

There are today numerous examples of systems formed from components that are themselves recognized as highly complex system with and without geographic distribution. Consider once again integrated air defense systems. If the integrated system has a single, strong central acquisition authority and operational command, it does not matter if the system is confined to a single ship or spread over hundreds of square miles. Conversely, if the system is formed of independently developed systems, which retain aspects of independent command, it does not matter if the integrated system is spread widely or is all resident on a single ship.

The only impact of geographic distribution is to limit the nature of interfaces between separated components. It is possible to tightly couple colocated components, and to more easily provide high bandwidth and low delay communication links. Colocated components can have power and material interfaces as well as information interfaces. The information interfaces can also be faster and have less delay. However, the only impact of this is to enable the construction of some kinds of tightly coupled system when they are not geographically distributed.

2.5. Design Heuristics and Examples of Systems-of-Systems

In the following sections three examples of existing and emergent systems are examined with respect to the two proposed criteria and the design principles. In reviewing the examples the reader should consider how the criteria impact how the system is built and evolved. A second part of the taxonomic argument, the consequences of misclassification, is taken up after.

Integrated Air Defense: The air defenses of modern military forces are commonly considered to be examples of systems-of-systems. An integrated air defense system is composed of a geographically dispersed network of semiautonomous elements. These include surveillance radars, passive surveillance systems, missile launch batteries, missile tracking and control sites, airborne surveillance and tracking radars, fighter aircraft, and antiaircraft artillery. All units are tied together by a communications network with command and control applied at local, regional, and national centers.

When operating as an integrated system, the network can exhibit networkwide emergent behavior. For example, optimized missile firing and engagement strategies and selective radar use to make targeting of individual elements difficult. However, most such systems are designed to be able to effectively fall back to less integrated configurations, and to make such transitions suddenly and in the heat of battle. Table I compares the proposed discriminating factors with the characteristics of integrated air defense systems and the use of the principles. The classification heading is dealt with in the main section to follow and is an additional taxonomic distinction.

The Internet: The Internet, the global computer-to-computer network, is an example of a collaborative system-of-systems. Its elements are themselves computer networks and major computer sites. Some of these component networks may also be composed of further subnetworks. Internet component sites collaboratively exchange information using documented protocols. Protocol adherence is largely voluntary with no central authority with coercive power. Coercive power emerges through agreements among major sites to block traffic and sites observed to misbehave.

Development, management, and operation are a collaborative effort among the participants. The principal coercive enforcement mechanism is the ability of the major service providers to refuse to carry noncompliant traffic. While there are a very large number of individual users, the number of backbone providers is fairly small, and almost all use equipment from a very small set of manufacturers. Thus the central collaborative group is small, and others are induced to follow their lead to take advantage of the backbone.

During earlier development, the Internet was controlled more directly by the U.S. government. Much prototyping is financed by the government. Also, during the earlier phases a considerable degree of technical control was exercised by a very small group (primarily two people) who provided architectural direction.

Table I. Integrated Air Defense and System-of-Systems Properties

Discriminating Factor	Applicability
Managerial independence of the elements	Component systems are acquired by separate program offices and run by separate operation units, sometimes in different military services. They are connected by their common membership in a military command structure
Operational independence of the elements	Connected by a military command and control network, which is integrating in both the technical and social sense. Each component is granted limited operational independence to respond to unforeseen and uncontrolled events
	Use of Design Principles
Stable intermediate forms	A variety of stable forms, both in time and space, are explicit in the design. Stable intermediates in operation are essential to combat robustness
Policy triage	Single service systems are centrally directed, but must deal with legacy equipment and politics. Multiservice systems concentrate on interfacing existing systems acquired in traditional service models. Some attempts to form more centrally directed multiservice systems
Leverage at the interfaces	Multiservice systems concentrate on information transfer. Single service systems also trade performance among components
Ensuring collaboration	Largely achieved through sociotechnical methods of command and control
	Classification
Directed	The system is developed and operated to a common purpose, and that common purpose is expressed through formal organizations, technical standards, and the socialization of its operators ("Boot Camp") to the common purpose

The Internet exhibits a rich set of emergent behaviors represented by the complex distributed applications that run on top of the communication substrate. Many of these were unanticipated at the time of original development, and many have evolved in unexpected ways. Thus new systems-of-systems have grown on top of the Internet's system-of-systems. The most complex of these is the World Wide Web, itself a system-of-systems that exists solely at upper protocol layers. The World Wide Web was planned for the exchange of scientific data, but is now used for diverse purposes including commercial, political, and illegal.

The Internet technical oversight group, the IETF, has had to carefully choose its standards. It has had to avoid putting large efforts into developing standards or extensions that could be implemented only if a central authority financed or dictated their use. Their approach has been to try to validate and standardize those approaches which have developed a consensus through use, and proactively establish standards that would then be the least cost option in emerging function areas.

Table II compares Internet characteristics and the proposed discriminating factors.

Intelligent Transport Systems: Intelligent Transport Systems (ITS) covers a wide range of potential applications of information and computer technology to road and transport networks. These range from improved public service vehicle communication to automated highways with robotically driven cars. As an example here consider only the portions of ITS generally known as Advanced Traveler Information Services (ATIS) and Advanced Traffic Control Systems (ATCS) and their fusion [IVHS America, 1992].

The goal of ATIS is to provide real-time information on traffic conditions and transportation options to travelers in any location. ATIS systems could allow a traveler to scan traffic conditions and choose the transportation mode with predicted least travel time. They could also allow a driver to get real-time traffic state and adapt her driving route accordingly.

The goal of ATCS is to allow a wide range of traffic control methods to be applied across metropolitan areas using strategies optimized from the information available. The information used could include real-time and predictive estimates of link times throughout the traffic network, and could include real-time statistics on driver start-destination points and planned route.

ATIS/ATCS fusion yields a very large, collaborative system-of-systems. Component fusion requires communication standards to allow interpretable data exchange. Building a fused system that works will require understanding the incentives needed for collaboration.

The structure or architecture of ITS is sketched in Figure 2. Loose boundaries have been drawn on the

Table II. The Internet and System-of-Systems Properties

Discriminating Factor	Applicability
Managerial independence of the elements	Component systems as acquired and operated by independent users. Component systems are developed (largely) by commercial firms following market dictates
Operational independence of the elements	Operational coordination is through voluntary adherence to technical standards. The standard setting process is also voluntary. The systems defense against noncooperators is only to exclude them. In the Internet's earlier stages of development it was more deliberately run by the U.S Government. Government sponsored projects continue to be important to the Internet's development
	User of Design Principles
Stable intermediate forms	The structure of the Internet is dynamic, with nodes being added and removed continuously and on their own volition. The main protocols are designed to allow evolution through replacement. The core protocol, IP, is now at version 4 with migration to version 6 beginning
Policy triage	The oversight bodies exercise very limited control, and carefully restrict their control to the network. Applications and underlying physical interconnects are controlled separately, if at all
Leverage at the interfaces	The architecture of the Internet is its interfaces. Nothing else is constant
Ensuring collaboration	The system fosters collaboration through low entry costs and benefits to cooperation. However, it is much weaker at excluding deliberate noncooperators, to the detriment of the system. This is a byproduct of its original development environment
	Classification
Collaborative	The system began with a directed purpose, but now follows purposes imposed upon it by its users. Operation and development is through the collaboration (largely voluntary) of its participants

figure to emphasize that portions of the overall system are broken across administrative and political as well as technical boundaries.

The boundaries are not unique, but represent one architectural choice about controlled versus collaborative operation [Maier, 1997; Lo, Hickman, and Weissenberger, 1995]. The purchase of vehicles with advanced intelligence will probably continue to be a primarily private transaction between individuals and corporations. The provision of data on destinations, positions, routes, and traffic state by private vehicles will probably be voluntary. On the other hand, highway network control will probably continue to be a public responsibility managed by politically chosen organizations. For the overall system to work well, not only must the technical components interface successfully, but the broader interaction of private choice and public policy must do as well, and it must do so compatibly with the technological architecture.

One can imagine ITS systems that do not meet the two criteria. It would be possible to build a centrally acquired and managed ITS, and there might be important benefits in doing so. If one could achieve the required social collaboration, a wide variety of traffic spreading and route optimization strategies could be implemented. As a demonstration of the linkage of the two criteria, consider the issues raised in attempting to design ITS as a system. If social collaboration does not occur as planned, the resulting system may be useless or worse than no system at all. Given the uncertainty and the political reality of defused authority, is there any possibility that a successful ITS could be designed that would take criteria two (managerial independence, simply because of political reality) but not criteria one (operational independence)? No such architecture has been proposed. Current architecture efforts focus on collaborative mechanisms. Central collaboration brokers exist [IVHS America, 1991] and directed planning of the architecture is being done [USDOT, 1994]. Table III again compares the characteristics of ITS with the proposed classification criteria.

2.6. Taxonomy: Virtual, Voluntary, and Directed Systems

Having identified the two criteria for systems-of-systems, it is natural to consider whether or not additional

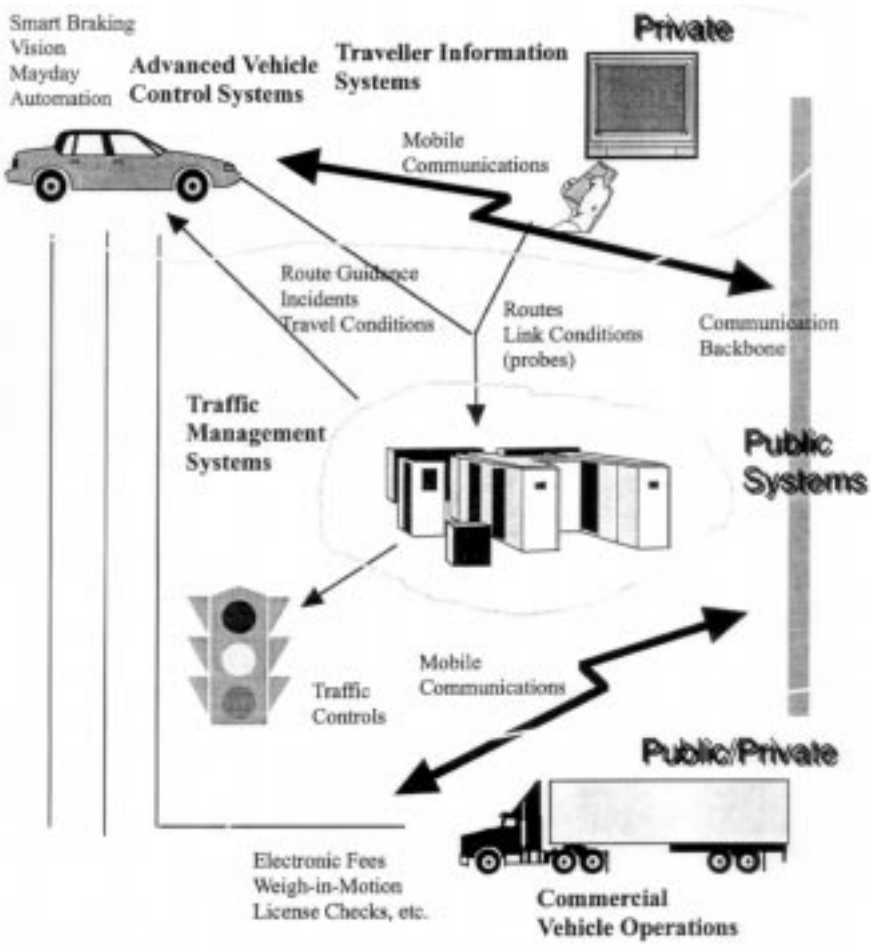

Figure 2 General structure of ITS showing technical and political boundaries.

Table III. Intelligent Transport Systems and System-of-Systems Properties

Discriminating Factor	Applicability
Managerial independence of the elements	Component systems are acquired and operated by independent users. Component systems are developed (largely) by commercial firms following market dictates
Operational independence of the elements	Operation will be through a complex mixture of individual and government action. Some components (traffic controls) will be run by public agencies at various levels. Other components will be run by private firms. All will require individual voluntary action by travelers
	Use of Design Principles
Stable intermediate forms	Since the system has not yet been built, adherence to design principles cannot yet be evaluated. A pervious paper has discussed the application of these design principles to Intelligent Transport Systems [Maier, 1997]
Policy triage	
Leverage at the interfaces	
Ensuring collaboration	
	Classification
Collaborative/virtual	No current body, voluntary or otherwise, control ITS related standards in the USA. Participants (governments, firms, users) will often have conflicting purposes which they will simultaneously attempt to fulfill

taxonomic distinctions exist. On this issue the evidence is much less clear. There appear to be three basic categories of systems-of-systems, distinguished by the form of managerial control. These distinctions appear to have discriminatory power, as discussed in the following section on misclassification.

Directed: Directed systems-of-systems are those in which the integrated system-of-systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes, and any new ones the system owners may wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose. For example, most integrated air defense networks are centrally managed to defend a region against enemy systems, although its component systems retain the ability to operate independently, and do so when needed under the stress of combat.

Collaborative: Collaborative systems-of-systems are distinct from directed systems in that the central management organization does not have coercive power to run the system. The component systems must, more or less, voluntarily collaborate to fulfill the agreed upon central purposes. The Internet is a collaborative system. The IETF works out standards, but has no power to enforce them. Agreements among the central players on service provision and rejection provide what enforcement mechanism there is to maintain standards. The Internet began as a directed system, controlled by the US Advanced Research Projects Agency, to share computer resources. Over time it has evolved from central control through unplanned collaborative mechanisms.

Virtual: Virtual systems-of-systems lack both a central management authority and centrally agreed upon purposes. Large-scale behavior emerges, and may be desirable, but the supersystem must rely upon relatively invisible mechanisms to maintain it.

A virtual system may be deliberate or accidental. Some examples are the current form of the World Wide Web and national economies. Both "systems" are distributed physically and managerially. The World Wide Web is even more distributed than the Internet in that no agency ever exerted direct central control, except at the earliest stages. Control has been exerted only through the publication of standards for resource naming, navigation, and document structure. Web sites choose to obey the standards or not at their own discretion. The system is controlled by the forces that make cooperation and compliance to the core standards desirable. The standards do not evolve in a controlled way, rather they emerge from the market success of various innovators. Moreover, the purposes the system fulfills are dynamic and change at the whim of the users. There have been recent efforts to create more centralized collaborative bodies (such as the WWW Consortium) to manage evolution.

National economies can be thought of as virtual systems. There are conscious attempts to architect these systems, through politics, but the long-term nature is determined by highly distributed, partially invisible mechanisms. The purposes expressed by the system emerge only through the collective actions of the systems participants.

2.7. Misclassification

Another test of the validity of the proposed classification is whether or not misclassification has any impact. Two general types of misclassification are possible. One is to incorrectly regard a system-of-systems as a monolithic system, or the reverse. Another is to misclassify a system-of-systems as directed, collaborative, or virtual.

Returning to the first case, system versus system-of-systems, consider the International Space Station (ISS). Is the ISS a system or system-of-systems? It appears to fit in between the criteria, and this matching to the criteria is one source of its problems. Its components are being developed with a large degree of independence (since they are being developed by different national space agencies), but they are very closely coupled in operation. In most cases components can fulfill very limited purposes, if any, independent of the other elements. There is very little redundancy of major functions across components. For example, one particular Russian component is required to keep the assemblage in orbit. Lacking that component none of the others can be flown. But the overall development organization has limited influence on the Russian national space agency to ensure that their critical component is delivered on schedule, or even at all.

In many respects the ISS is a collaborative system. The integrating organization (NASA) has considerable authority over purposes and component specifications and interaction. But the international distribution of component development breaks the unity of decision over purpose and behavior. These conflicts have to be resolved collaboratively. However, the close coupling of the components severely limits the ability to sever any part or evolve along very different paths.

One example of such a mechanism is the heuristic on stability. Applied to the space station, it would mean the station should be technically and operationally self-supporting with any component severed from the configuration. This leads to considerable redundancy, and higher directly measurable costs. However, it also leads

to greater robustness in development, which might well lower costs. Implemented another way, the heuristic suggests seeking a configuration in which each participant gets greater benefits (evaluated on their own terms) by participating than by going their own way. A failure to incorporate such mechanisms may lead to instability in configuration or operation.

The cost and benefit issues of making the ISS a system versus a system-of-systems are fairly clear, although making the tradeoff precise would be quite difficult. A system architecture that met both criteria would clearly be more expensive than one that does not. It would require redundant provision of major functions (like propulsion and life support) so that multiple subconfigurations were operationally stable. If the management perspective is to minimize cost, with the assumption of management control, then architecting as a system rather than a system-of-systems is clearly called for. If one assumes that the requisite control does not exist, then a redundant architecture is called for. The system-of-systems version of a space station would be inherently evolutionary, and would only have target configurations. The architecture would be of the interfaces between major blocks, with those interfaces designed to allow units to be added and subtracted while maintaining operations. Moreover, the initial building blocks would be relatively self-contained to allow stable operation with very few blocks delivered.

For the second case, classification within the system-of-systems taxonomy, consider a multiservice integrated battle management system. Military C4I systems are normally thought of as directed systems-of-systems. As the levels of integration cross higher and higher administrative boundaries the ability to centrally control the acquisition and operation of the system lessen. In a multiservice battle management system there is likely to be much weaker central control across service boundaries then within those boundaries. A mechanism that ensures components will collaborate within a single service's system-of-systems, say a set of command operational procedures, may be insufficient across services.

In general, if a collaborative system-of-systems is misclassified as directed, the builders and operators will have less control over purpose and operation than they may believe. They may use inappropriate mechanisms for insuring collaboration and may assume cooperative operations across administrative boundaries that will not reliably occur in practice. The designer of a directed system-of-systems can require that an element behave in a fashion not to its own advantage (at least to an extent). In a collaborative system-of-systems, it is unlikely that a component will be induced to behave to its own detriment.

A virtual system-of-systems misclassified as collaborative may show very unexpected emergent behaviors. In a virtual system-of-systems neither the purpose nor structure are under direct control, even of a collaborative body. Hence new purposes and corresponding behaviors may arise at any time. The large scale distributed applications on the Internet, for example USENET and the World Wide Web, exhibit this. Both were originally intended for exchange of research information in a collaborative environment, but are now used for diverse communication purposes, including undesired and even illegal purposes.

One of the design heuristics for systems-of-systems is to seek stable intermediate forms. The heuristic suggests system architectures must possess stable forms, both technical and political. One aspect of stability is the ability to sever any portion of the system and continue operation. In a directed system the stability of the form can be assured by the stability of decisions in the controlling body. In a collaborative system stability is achieved only through the interaction of the underlying preferences of the participants. As is well known, many multiple stakeholder decision-making methods are not stable in the sense that they do not produce transitive preferences.

In contrast, misclassification of the complexity of system components or their geographic distributions (alternative system-of-systems criteria from other authors) has detailed technical consequences rather than these architecture level consequences. Understanding the complexity level of components is needed for reliability and cost modeling, but effects the overall architecture much less. The geographic distribution will effect strongly the technical nature of communication, but becomes dominant when components must be linked with very short response times.

2.8. Communications as Architecture

To summarize the argument so far, we propose two criteria for classifying a system as a system-of-systems. The two criteria are that the system be severable into components which can continue to operate to fulfill their own purposes, and that the components continue to operate (at least in part) to fulfill their own purposes even after integration into the system-of-systems. Several design heuristics are also proposed. The net effect of these heuristics is that the system-of-systems architect should concentrate on interfaces and how they foster or discourage collaborative emergent functions in the system-of-systems. The principles of "leverage at the interfaces," "policy triage," "stable intermediate forms," and "ensuring collaboration" combine to a fo-

cus on communications as architecture. The conclusions to be drawn from each are:

From leverage at the interfaces we conclude that interfaces are the architecture. If components are procured semi-independently then the *standards* of communication are more important than any particular component system.

From policy triage we conclude that not everything can be standardized or defined. The points of leverage must be discerned and the architect's resources applied sparingly.

From stable intermediate forms we conclude that the interfaces must support severability in either vertical or horizontal directions. Vertical severability means the ability to remove or add a physical component to the system-of-systems. Horizontal severability means the ability to add or remove applications or functions to the system-of-systems independently of the physical components.

From ensuring collaboration we conclude that attention must be paid to how the participating components derive value from participation.

Returning to the examples, it is apparent that the architecture of each is defined through communications. If more than information is exchanged similar issues appear. The issue generalizes to interface standards rather than just communication standards. The following discussion develops the conclusions from the heuristics in greater specificity to the communications standard domain.

Communications standards are commonly defined in terms of a layered communication model. The layered model divides the communication process into a stack. Each component of the stack is referred to as a layer. The reference model for communication system layering has been the seven-layer OSI model [Tannenbaum, 1989]. Following more recent practice, a better model may be to consider five layers [Tannenbaum, 1995].

The application layer. The user level application processing.

Upper layer(s): Object standards, global naming, standards for semantic content in user to user message passing. Sometimes known as "middleware."

Transport: End-point to end-point arbitrary message transfer.

Network: End-point to End-point unreliable single packet transfer with an upper bound on packet size. Convergence on IPv6 is likely in the future over a broad range of applications.

Physical, Media and Data Link Layers: Point to point data transfer including low level reliability, contention and access control, and modulation issues.

The structure of a five-layer model is shown schematically in Figure 3. The traditional seven-layer model divides the bottom layer into two, and calls out two specific upper layers, the session and presentation layers. In practice, the choice of physical and/or data link layer is becoming less significant. A wide variety of bit level transport media are available, and it is rarely desirable to design a new one for a specific application. Hence, the architecture of communication for a system-of-systems is more likely to concentrate on the layers above bit transfer to focus on the areas unique to the system. Bit transfer will usually be provided by the emerging backbone of wide area communication services.

The upper layer situation is less well defined. Neither of the OSI model's two upper intermediate layers (session and presentation) have seen wide implementation. Moreover, the communication abstractions suggested by those layers do not seem to match well the actual structures of computer and convergent communication. Instead a variety of upper layer intermediate toolkits have appeared. Many of the newer ones are based on object-oriented abstractions of interprogram communication or other models of computer-to-computer communication [Next, 1996; Open Software Foundation, 1996].

A communication standard may encompass any set of layers. Standards typically cover only a single layer, but an integrated set of multilayer standards may be needed for a particular application. The example systems contain a diversity of cases. Some of the most widely known standards are those for the network and transport layers, such as TCP/IP and IPv6 [Comer, 1995], SPX/IPX and AppleTalk [Sidhu, 1990]. Asynchronous Transfer Mode (ATM) largely fits into this category.

Going back to the examples, each shows that its architecture, in the sense of fundamental or unifying structure, is largely defined by a communications model.

Network Layer: The Internet

The Internet is an existing system-of-systems, and it is built on standards. In particular, it is built on a critical standard, the Internet Protocol (IP). This standard—several standards, actually—defines the structure of a data packet, globally routable addressing, routing methods, and an internode control protocol. In addition, standards exist for mapping IP onto various lower layers

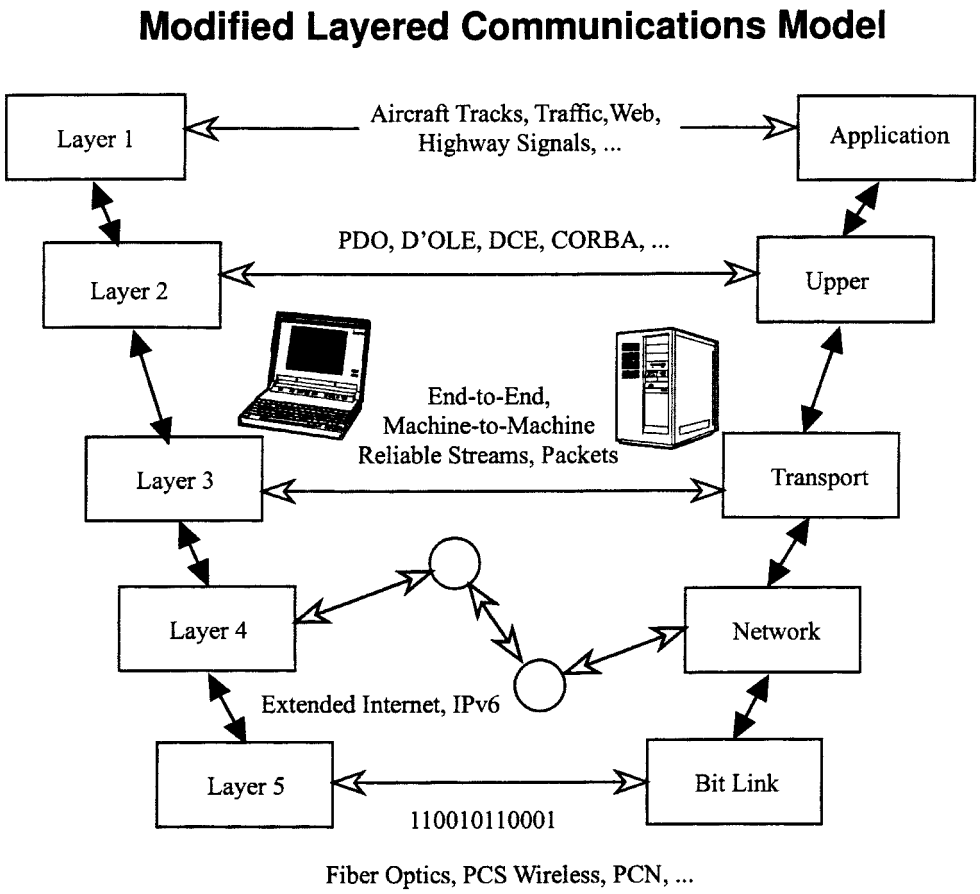

Figure 3 A five layer model of communications based on existing and emerging Internet concepts.

for the diversity of physical interconnections, and for various higher layers that operate on top of IP. The situation was illustrated in . While IP itself is all that is necessary for Internet data exchange to exist, useful applications, the emergent behaviors observed by the users, require higher layer standards from which the emergent applications are built.

Physical Layer Up: ITS Beacon System

In a system as diverse as the ITS, it is not surprising to find a diversity of communication systems and standards. In many cases key ITS architectural elements can be defined as communication standards in the upper layers only [Maier, 1997; Kady and Ristenbatt, 1993]. But in one particular case there is a potential requirement for an ITS unique communication system that would include standards from the physical layer up. This case is the short-range vehicle beacon system. The beacon system is envisioned as an infrared (or possibly microwave) based system that will communicate between vehicles and roadside transceivers over distances of a few meters to tens of meters. This system has several important attributes:

- Because it is short range and directed it provides an enormous aggregate bandwidth for communications over the entire vehicle population of a metropolitan area. It could support near real-time independent interaction with every vehicle if so desired.
- Since roadside beacon locations are known, it combines communication and position determination and reporting. It does so at low cost since it combines the functions and the unit itself is potentially very low cost.
- It could be used for intervehicle communication and cooperative sensing as well, paving the way for automated highway operation.

All of these attributes require a standard for the system from the physical layer up. Commonalty over a nation's road network is required in all of modulation, wavelength, data linking, and message content.

Upper Layers: ITS Information Exchange

ATIS and ATCS fusion in ITS does not require a beacon system. It does require a set of standards at higher layers. These standards form fundamental and unifying structures that cross-multiple ITS implementations and multiple evolutionary stages. If one assumes that Internet-like communications will be ubiquitous for computer-to-computer nodes in the ITS time frame (the next twenty years), several not-now-existing standards are relevant to ITS. The author has previously argued [Maier, 1997] that the architecture of an ITS is defined by standards for:

Geographic referencing. When a message says "I'm going from here to there," how do we define here and there so all receivers understand? This type of messaging requires a standard means of reporting location and correlation to maps.

Traffic message content. The ITS needs messages that report traffic state. Such messages must include location (already discussed above), but must also include state. In general the ITS needs a standard for message types and contents that map to underlying transport mechanisms.

The main emerging standards for upper intermediate layers are in computer-to-computer communication. Among well-known examples are Enterprise Java, Microsoft's Distributed Common Object Model (DCOM) and its progeny, and the Distributed Computing Environment (DCE). It is likely that these will become the building blocks for domain specific upper layer standards, which will define the architectures of some systems-of-systems.

Eclectic: Integrated Air Defense

Modern military systems have an eclectic array of communication systems and standards. Because of the long life of many military systems, old communication interfaces may be maintained long after their technology has become obsolete. This diversity of communications is often a burden to the military architect. The existence of this burden is a testimony to the importance of elegant and insightful communication standards in systems-of-systems.

This eclectic mix defines the architecture of military systems-of-systems. Among the mix will be communication standards similar to those discussed previously. These include low level data link standards, position referencing standards, and message syntactic and semantic content standards. But an integrated air defense system will have other interface standards that are different in nature from those discussed in the other examples. An example is command and control, or operational, standards.

Command and control or operational standards define how application specific information is to be generated and passed, and how each element is to determine its own actions with the framework of the system-of-systems. These standards are implemented technically (in the component designs), in documented operational standards, and in the shared assumptions of the operators. The operators shared assumptions are deliberately produced through the training process.

This eclectic and evolutionary mix is addressed through training and layering. An insightful standard on one layer can fruitfully live on long past the obsolescence of the physical layers on which it originally ran. For example, the IP standard uses a version numbering system to allow packets from different versions of IP to coexist on the Internet. This ability has been important in allowing the Internet to evolve, and will be used again during the evolution to IPv6 from the current IPv4.

If the architecture is in upper layer communication standards, the architect must be concerned about representation models for these layers. Unfortunately, this is not a well-developed modeling field. Upper layer communications are dominantly software object to software object. Object-oriented modeling methods [Booch, 1995] provide one means of representing these communication structures. Current work on software architecture representation at the object level does not include a rich syntax for high level communications [Shaw, 1996]. The communication elements are targeted at a somewhat lower level. However, communication representations are extensively treated in domain specific software architectures [Hilliard, 1990]. Some recognition of these issues can be found in definitions of communication-centric software architectures [Boasson, 1996].

3. CONCLUSIONS

The basic thrust of this paper has been the study of a taxonomic grouping of systems. To be useful, a taxonomic grouping must have discriminatory value. It must distinguish between groups of systems that differ in the characteristics of good practices in development. This paper has argued that systems that meet the two-fold test of operational and managerial independence of the components form an important taxonomic grouping. This grouping encompasses many of those systems commonly termed "systems-of-systems," and provides a useful taxonomic grouping that might logically be called "systems-of-systems." Collaborative systems

might be a better term, as it much more clearly expresses the nature of the grouping.

By examining some examples of collaborative systems some heuristics for success become clear. The heuristics are not themselves unique to collaborative systems, rather they are refinements of more general heuristics that have appeared elsewhere [Rechtin, 1991; Maier, 1994]. It is the refinements that are interesting since they gain prescriptive force by the incorporation of domain knowledge. The design and process recommendations center on four refined heuristics, communications as architecture, and the problems of misclassification. The four refined heuristics produced the recommendations:

1. Stable Intermediate Forms: A collaborative system designer must pay closer attention to the intermediate steps in a planned evolution. The collaborative system will take on intermediate forms *dynamically* and *without direction*, as part of its nature. Thus, careful attention must be paid to the existence and stability (in all suitable dimensions) of partial assemblages of components.
2. Policy Triage: The collaborative system designer will not have coercive control over the systems configuration and evolution. This makes choosing the points at which to influence the design more important. In communication-centric systems, this means that design leverage will frequently be found in relatively abstract components (like data standards and network protocols).
3. Leverage at the Interfaces: A collaborative system is defined by its interfaces. The interfaces, whether thought of as the actual physical interconnections or as higher level service abstractions, are the primary points at which the designer can exert control.
4. Ensuring Cooperation: A collaborative system exists because the partially independent elements decide to collaborate. The designer must consider why they will choose to collaborate and foster those reasons in the design. This is not a consideration in the design of monolithic systems where the components can operate only as part of the whole.

The overarching consideration is architecture as communications. In a collaborative system (a system-of-systems) the intersystem communications is the architecture (in the sense of the organizing structure). Thus system-of-systems architecting is largely an exercise in communications architecting. In the current technological environment this usually does not equate to design attention to the physical layer of bit transfer. Instead the primary considerations are likely to be found at higher layers in a network protocol stack, especially at the network layers and any middleware layers between the applications of interest and the transport layer.

Examining misclassification is one method for evaluating the value of the proposed taxonomic grouping. Misclassified systems have characteristic problems in design, development, and use. In a similar vein, Shenhar [1994] identified distinct problems of misclassification in his taxonomic proposals. The fundamental error in misclassification is treating a collaborative system as if it were a monolithic system. The designers, believing that they have control where they do not, will be motivated to remove redundancy and stable intermediate forms in the interest of lowering costs. However, a collaborative system actually assumes a configuration that represents a collaborative equilibrium. In the absence of appropriate redundancy or intermediates the equilibrium may be no system at all as the components choose not to participate.

It would be desirable to test the proposed heuristics in a broader way through detailed case study. As in most systems engineering studies, formal experiment is not really possible. We don't build duplicate complex systems by different methods just to see what would happen. We can look retrospectively at built systems to test the applicability of heuristics, however. It is in that spirit that the analysis given here is offered.

Whether the most appropriate term is system-of-system, federated system, or collaborative system; this paper argues that the most appropriate taxonomic node is for systems which consist of semiindependent, collaborative components. These systems-of-systems are the children of modern communications and computing. They will exist more widely in the future as individual systems become "smarter" and communication interfaces become routine. Architecting and engineering them will not be a simple repeat of how systems have been architected and engineered. The nature of the communication standards that enable and define individual systems-of-systems should shift from physical-up standards to higher layer standards that assume the existence of an IP-like data transport substrate; and much greater attention must be paid to how components develop and maintain collaborative relationships.

Collaborative and virtual systems-of-systems will also become more common with the ubiquity of smart systems independently operated and managed. This will place a premium on the discovery and clever use of design principles that produce emergent behavior through voluntary collaboration. A fruitful area for such

work may be in the use of pseudo-economic mechanisms.

REFERENCES

Bertsekas, D., and Gallager, R., *Data Networks*, Prentice Hall, Englewood Cliffs, NJ, 1992.

Boasson, M., "Subscription as a Model for the Architecture of Embedded Systems," *Proc. 2nd Annu. Conf. Eng. Complex Comput. Syst.*, 1996, pp. 130–133.

Booch, G., and Rumbaugh, J., *Unified Method for Object-Oriented Development*, Rational Corporation, 1995.

Brock, W. A., and Durlauf, S. N., "Discrete Choice with Social Interactions I: Theory," Santa Fe Institute Working Paper 95-10-084, 1995.

Butler, S., Diskin, D., Howes, N., and Jordan, K., "The Architectural Design of the Common Operating Environment for the Global Command and Control System," *IEEE Software*, November 1996, pp.57–66.

Chiariglione, L., "Impact of MPEG Standards on Multimedia Industry," *IEEE Proceedings*, Vol. 86, No. 6, June 1998, pp. 1222–1227.

Comer, D. E., *Internetworking with TCP/IP*, Vol. 1, 3rd ed., Prentice Hall, Englewood Cliffs, NJ, 1995.

Eisner, H., "RCASSE: Rapid Computer-Aided Systems of Systems (S2) Engineering," *Proc. 3rd Int. Symp. Natl. Council Syst. Eng.*, NCOSE, Vol. 1, 1993, pp. 267–273.

Hayes, R. H., Wheelwright, S. C., and Clark, K. B., *Dynamic Manufacturing* The Free Press, A Division of Macmillan, New York, 1988

Hilliard, R. F., "The Notion of 'Architecture' in Model-Based Software Engineering," *Proc. Workshop Domain-Specific Architectures*, Hidden Valley, PA, July 1990.

IVHS America, System Architecture Committee, IVHS Goals and Objectives, , Washington, DC, 1991.

IVHS America, Strategic Plan for Intelligent Vehicle-Highway Systems in the United States, IVHS America, Report No. IVHS-AMER-92-3, Intelligent Vehicle-Highway Society of America, Washington, DC, 1992.

Kady, M. A., and Ristenbatt, M. P., "An Evolutionary IVHS Communication Architecture," *Proc. 1993 Vehicle Navigation Inf. Syst. Conf. VNIS '93*, 1993, pp. 271–276.

Lo., H., Hickman, M., and Weissenberger, S, "A Structured Approach for ITS Architecture Representation and Evaluation," *Proc. 1995 Vehicle Navigation Inf. Syst. Conf. VNIS '95*, 1995, pp. 442–449.

Maier, M. W., "Heuristic Extrapolation in System Architecture," *Proc. 4th Int. Symp. Natl. Council Syst. Eng.*, NCOSE, Vol. 1, 1994, pp. 525–532.

Maier, M. W., "On Architecting and ITS," *Systems Engineering/IEEE Transactions on Aerospace and Electronic Systems*, AES Vol. 33, No. 2, April 1997, pp. 610–625.

NeXT Computer, "White Paper: Architecting for Change with Enterprise Objects Framework and Portable Distributed Objects," NeXT Computer White Paper, 1996.

Open Software Foundation, DCE Overview, OSF-DCE-PD-1090-4, 1996.

Peterson, L., and Davie, B., *Computer Networks: A Systems Approach*, Morgan-Kaufman, San Mateo, CA, 1996.

Rechtin, E., *System Architecting: Creating and Building Complex Systems*, Prentice Hall, Englewood Cliffs, NJ, 1991.

Rechtin, E. R., and Maier, M. W., *The Art of Systems Architecting*, CRC Press, Boca Raton, FL, 1997.

Sage, A., *Methodology for Large Scale Systems*, McGraw-Hill, New York, 1977.

Schuman, R., "Developing an Architecture that No One Owns: The U.S. Approach to System Architecture," *Proc. First World Cong. Appl. Transport Telematics Intelligent Vehicle-Highway Syst.*, Paris, France, 1994.

Shaw, M., and Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, Englewood Cliffs, NJ, 1996

Shenhar, A., "A New Systems Engineering Taxonomy," *Proc. 4th Int. Symp. Natl. Council Syst. Eng.*, National Council on System Engineering, Vol. 2, 1994, pp. 261–276.

Sidhu, G. S., *Inside AppleTalk*, 2nd ed., Addison Wesley, Reading, MA, 1990.

Stuart, J. R., "Teledesic Network and Space Infrastructure Architecture and Design Features," Proc. 2nd Annu. Conf. Eng. Complex Comput. Syst., 1996, pp. 147–150.

Tannenbaum, A. S., *Computer Networks*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 1989.

Tannenbaum, A. S., *Computer Networks*, 3rd ed., Prentice Hall, Englewood Cliffs, NJ, 1995.

USDOT, ITS Architecture Development Program Phase I: Summary Report, Washington, DC, November 1994.

USDOT, National Program Plan for ITS, Washington, DC, 1995.

Mark W. Maier received the B.S. and M.S. degrees from the California Institute of Technology and the Engineer and Ph.D. degrees in Electrical Engineering from the University of Southern California. While at USC, he held a Hughes Aircraft Company Doctoral Fellowship, where he was also employed as asection head. Currently, he is a Senior Engineering Specialist at the Aerospace Corporation in Chantilly, Virginia, and is also on leave from his position as Associate Professor of Electrical and Computer Engineering at the University of Alabama in Huntsville. Dr. Maier's research interests are in system architecting for large-scale systems, computer based system design, randomized radar waveforms, and data compression.