Multiplicity of Perspectives, Context Scope, and Context Shifting Events

Hans Polzer
Integrated Systems & Solutions
Lockheed Martin Corporation
Fairfax, VA, USA
hans.w.polzer@lmco.com

Daniel A. DeLaurentis
School of Aeronautics & Astronautics
Purdue University
West Lafayette, IN, USA
ddelaure@purdue.edu

Donald N. Fry
School of Aeronautics & Astronautics
Purdue University
West Lafayette, IN, USA
dfry@purdue.edu

Abstract—*A perspective is a system's (or individual's) version of operational context. This paper explores the issue of the multiplicity of perspectives. It is hypothesized that "trigger events" bring two or more such perspectives into novel contact with each other in a way that challenges the context and scope assumptions of the systems and their users. This disjunction of perspectives presents a root cause of interoperability issues at the systems of systems level. We propose the use of a structured description of a system of systems as a means to specifically articulate the trigger events. By making context and scope assumptions explicitly visible and accessible, individual systems will be able to discover them and adapt to each other, enabling the dynamic creation of new, re-purposed, and re-scoped systems of systems. Two 'thought experiments' are offered to illustrate the concept of trigger events and their role in bringing perspectives into conflict.*

Keywords: system of systems, operational context, perspective, interoperability, trigger event.

1 Introduction

An *operational context* can be defined as "the interrelated conditions which exemplify a system's state of being and which describe its purpose, scope, and meaning for services it may offer." An operational context can change over time, and a particular system (or individual) can be in multiple operational contexts simultaneously. For example, a person may be working at home (normal work context aside from location) and waiting for the carpet installers to arrive (personal life, household task context). A related concept is that of *perspective*, which is "a system's (or individual's) version of operational context." Further, a system's *frame-of-reference* (FoR) brings "the institutional background and experiences that shapes perspective." Note that perspective is not just the property of an individual or role player (e.g., participant, stakeholder). It is also something a system has, based on the role the sponsors who created the system had in mind for it. The same applies to frame-of-reference.

Such perspectives persist independently of each other within different systems and users and allow rapid establishment of shared context with applications and other users for some session duration associated with user and institutional operational objectives. However, disparity in perspective and/or context is a major obstacle in dynamic reconfiguration of interoperating systems. Such obstacles to dynamic reconfiguration of interoperating systems—dynamic interoperability—lies at the heart of the challenge of fielding a system of systems.

A key finding of the recently released *Report on Systems-of-Systems Engineering for Air Force Capability Development* from the US Air Force Scientific Advisory Board (SAB) finds that "convergence protocol" are essential for future success in fielding system of systems [1]. These convergence protocols are meant, essentially, to resolve operational context differences, once detected—a sort of discovery and adaptation capability. However, convergence protocols/services are expensive, and humans operate very effectively based on implicit context sharing. There is a general aversion to passing information we know the other person/system already has. Hence, we want to invoke convergence protocols only when we feel it is cost-effective to do so (such as in an email exchange). Further, the issue remains of how to make the difference visible and discoverable in the first place. For example, how do we know when the scope assumptions that we are operating under are still valid, or when they are no longer valid? If they are built into a system at requirements and design time, but are not stated explicitly in the run-time system and validated periodically or continuously, we may be operating outside our scope assumptions without realizing it. This can and probably does lead to anomalous system behavior, both internally and with external systems.

The purpose of this paper is to couch perspectives in a system of system setting and explore the influence of events that bring multiple perspectives into contact—termed "trigger events." The nature of such influence depends on the institutional frames of reference and associated operational context scope that system users and system designers implicitly assume in their behavior and their designs. We provide two thought experiments in differing domains to make these ideas concrete. Our message in this presentation is that making context and scope assumptions visible is the prerequisite for executing convergence protocols. More broadly, making context and system perspectives overtly visible to system users is an important precondition for effective human system integration in a system of systems context.

1-4244-1160-2/07/$25.00 ©2007 IEEE.

2 Context/Scope Dimensionality

In a system of systems, operational context confusion can result in mismatches due to differences in level of awareness as well as differences across scope dimensions. Thus, an organizational structure and common language with naming conventions/authorities across disciplines, domains, and institutions is required. We adopt the model from the authors in Ref. [2] which includes technical, economic, operational, or political aspects at various hierarchical levels (see Figure 1). The categorization of the levels lends clarity to scope boundaries while maintaining the lucidity in abstraction provided by the levels. Representations of entities within a system of systems may vary across these dimensions and unless the context in which the system operates in each dimension is explicitly shared/understood, misrepresentation can result.

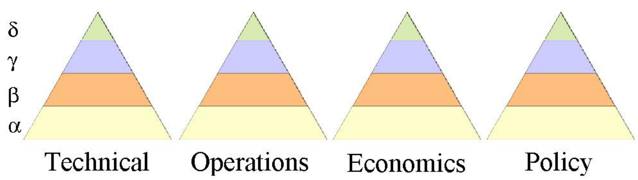

Figure 1. Scope dimensions and hierarchical levels of system of systems

Due to the scope of the problems SoS engineering take on, approaches such as those in Ref. [3] quickly become encumbered by the sheer number of perspectives involved. Further, these methods are primarily focused on requirements selection for system development, rather than the integration of new and existing systems into a system-of-systems.

3 Hypothesis

In complex situations, events unfold and circumstances change in ways that bring perspectives into contact with each other that were previously assumed to be independent or interrelated only in ways that did not anticipate the new event or circumstances. These "trigger events" bring perspectives into novel contact with each other and challenge the context and scope assumptions of the systems and their users. In other words, they motivate the creation of a new "system of systems" context not anticipated by the individual system sponsors and designers. This new context perspective exposes the mismatch in respective context and scope assumptions in the previously independent systems and institutions. Net-Centricity puts a premium on the ability to dynamically form new systems of systems in response to changing operational objectives and circumstances. Making context and scope assumptions explicitly visible and accessible over the network will enable and facilitate individual systems to discover them and to adapt to each other's context/scope, or allow third-party systems to function as composition/adaptation agents for dynamic creation of new, re-purposed, and re-scoped systems of systems.

Other sources of conflict include simply an oversight on the part of the system designers. In other words, the sponsors of the systems anticipated the differences but the designers failed to adequately account for all of them. This is addressed somewhat by the methods proposed by Finkelstein, *et al.* for integrating multiple perspectives in system development [4]. Furthermore, there may be hysteresis effects in the condition changes that require either a larger condition change or some major event to make it evident to the respective systems that their scope/context assumptions have been violated. In other words you may have to go some distance outside the box before you realize you have left the comfort of your box. In some ways, this may be the most dangerous situation—your system assumptions are no longer valid, but you do not yet realize that is the case, so you continue operating as if everything is still normal. In part, this situation may be due to the fact that scope and context boundaries along some dimensions are not clearly delineated—shades of grey, if you will. For example, product categories in some product catalog might overlap and a product might be in either or both categories, depending on the intended use. And the system may not have considered all possible uses of a product and thus did not categorize the product in the context of such possible uses. Another system or user looking for products that support such "unanticipated" uses of the product would not find the product unless it anticipates some mapping of its intended use to other product categories that the cataloging system does support.

More often, the undetected "out of context/scope" situation is due to the fact that the system does not overtly check for context/scope assumption violations. In the past, systems were built for some prescribed purpose by some sponsor who inherently understood what the system did and why it made the assumptions that it did. The advent of the Internet and associated net-centric concepts supporting the dynamic formation of systems of systems is challenging this paradigm. Because there is no general model for describing context and scope assumptions among systems, even systems built with the net-centric model in mind will check for only a few of the parameters/dimensions that might characterize a given system and its appropriateness for use in a novel context.

4 Experiments

Two 'thought experiments' are offered in this section to illustrate the potential for trigger events and their role in bringing perspectives into conflict. In each example, SoS scope categories and levels will be included parenthetically to tie the organizational structure into dealing with the interconnected systems. These examples also serve as foundation for problem statements that can be subsequently explored quantitatively using modeling and simulation. The purpose of this future modeling work would be to generate classes of events and characterize particular responses due to these events. Ultimately, a goal would be to develop mechanisms to detect potential conflicts as a precursor to the synthesis of convergence protocols.

4.1 Own-Force Awareness

One thought experiment we offer up here is that of systems developed and deployed to make it possible to know where one's force elements might be located at any given point in time. In the commercial world, this might be termed "resource" or "asset" awareness. In the military environment, own-force awareness could also include awareness of other attributes besides current location, such as current fuel status, ammunition levels, and the like.

Of course, the military has to be able to operate in locations where there is no available networking infrastructure. They also have security concerns that put a premium on not revealing their location to anyone other than authorized force elements. The solution typically includes some form of satellite communications path by which a vehicle-mounted transponder reports its current location periodically to a data collection point. The data from all the vehicles being managed is then aggregated into some reporting service that allows appropriate personnel (as defined by policy) to determine where individual vehicles may be, and thereby infer overall force makeup based on their area of operations or force element affiliation.

However, most such systems do not offer the ability to track individual soldiers, nor is there just one such system used for tracking all force elements. Different types of force elements have different location reporting systems, based on a variety of institutional scope boundaries, including force element functionality (e.g., logistics versus combat units), force element type (e.g., naval versus ground forces), area of operations, or force element nationality. That makes this a system of systems problem with multiple scope dimensions characterizing the domain and context of each system. Note also that the description of the scenario so far tends to lead one to an implicit scope assumption that all of the systems involved are military systems. But of course, force elements may be local police, contractors, leased construction equipment, or just about anything potentially useful for conducting operations in a given area (e.g., newspapers or newspaper delivery trucks). Most military systems, however, are designed to operate autonomously (for good reason) and generally are not expected to work with force elements that are not "organic," to use the military jargon (again, for good reasons). We will return to this point later in the thought experiment.

As the previous paragraph implied, there are multiple opportunities for disparate systems to have to work together in unanticipated ways. We will explore two of them in this paper, albeit relatively briefly due to space constraints.

Scenario 1—The first "trigger event" we will look at is that of a time-critical need arising to locate a particular soldier (a variant of the "Saving Private Ryan" scenario). This type of trigger event is at the α level in our SoS model, and primarily in the operational scope domain, although it also has some economic and policy domain aspects, as we shall see. Let's say that we need to find one of our soldiers who has a particular language or technical skill to help with interrogating someone who has just been captured. This creates two potential unanticipated system of systems conflict interactions.

The first is finding soldiers with the requisite skills in our own-force management system. Typically skills that are not integral to a soldier's role in a force element are not managed by own-force awareness systems. This type of data is usually the province of personnel and training management systems. Furthermore, these systems have an individual soldier perspective rather than a force element perspective. They might include the soldier's current unit assignment by unit name or ID, but probably not as a retrieval key. So, for example, it is usually not easy to query such a system based on the current unit individuals are assigned to as a constraint because that is not the normal way a personnel skills system is used. Even if it has this capability, such a system almost certainly does not know a given unit's location. So how would someone find soldiers near some location that might have the requisite skills?

One could start with the set of all soldiers with the requisite skills in all force elements from the skills system. Or one could start with all the soldiers within some distance of the location of need in the own-force awareness system. But the latter case presents yet another problem. The location awareness of the own-force awareness system only extends to vehicles, not individual soldiers. Ideally, the own-force awareness system would include a mapping of which soldiers are in which vehicles. In actuality this is unlikely because for most force employment purposes the location of specific soldiers is usually not an important attribute to track at the force level, and doing so would make such systems more expensive (economics) and potentially make every soldier more detectable by an adversary (policy/doctrine). The upshot is that we need to retrieve the identity of all vehicles within some radius of the need location, and then retrieve the identity of all soldiers in those vehicles. But this begs the questions of whether all soldiers are actually assigned to vehicles, when in reality some may be operating dismounted. An alternative approach is to use some aggregate of all the vehicle locations for vehicles assigned to specific force elements in an area of operations as the general location for each of the associated force elements, and then infer that all soldiers assigned to that force element are in that general location.

While this is probably not very precise from a location perspective, it is often the best approximation to finding soldiers within some distance of the need location. Given this information, we still need to find which of the available soldiers have the requisite skills. Here we hit another perspective clash as we try to access the skills system using a specific set of soldier identities. The first potential problem is whether the skills system uses the same soldier identifiers as the own-force awareness system (a representation issue). For example, the skills system might use social security number as the primary identifier because it is part of the personnel management domain, while the own-force awareness system might use soldier name due to privacy concerns (policy). Although this is not a trivial problem because of the many possible spelling and duplicate name issues (technical, to some degree), we'll grant that the name matching will work. However, it is unlikely that the skills system is designed to accept batches of names and a skill category as input parameters, again because that is not the normal way such a system is used (perspective issue). Instead it is likely that the skills system will require each candidate name to be submitted individually via some query service interface and return a personnel record which will include all the skill codes associated with that soldier.

The alternative path of starting with the skills system has similar perspective clash problems. It should be easy to use the skills system to retrieve a list of candidate soldiers with the requisite skills, because that fits within the perspective of such a system. Qualifying this request by specific force elements is much less likely to be supported because that does not fall within a normal personnel management perspective. But even if it is, it is unlikely that the skills system is aware of current force element locations. And it is equally unlikely that the skills system was designed to access the own-force awareness system to obtain force element location information. We therefore feel confident that the system of systems context that was generated by the "Saving Private Ryan" trigger event creates a perspective clash unlikely to be resolved by these systems as they are designed today. In other words, these systems would be called non-interoperable by the press if it got wind of the situation. Human intervention would be required to make this system of systems context work.

It is important to note that this trigger event was applied in the context of systems whose scope is the same military Service, such as an Army for a single country. If the force elements were multi-Service, multi-functional domain, or even multi-national, these perspective clashes would be much more severe, and there would be a lot more systems involved in them. To illustrate this point, let's look at another possible trigger event in the own-force awareness system context.

Scenario 2—A major disaster relief operation is underway in a foreign country. The force elements from a different country have been airlifted into the disaster area and have started to deploy throughout the area with the vehicles they brought with them. This allows them to deploy their own-force awareness system. Note that this trigger event is a δ or γ-level event with both operational and economic scope dimensions. There is a policy scope aspect to this as well because of possible cultural and legal differences between the country experiencing the disaster, the assisting country, and possible third party elements such as the International Red Cross.

In this system of systems context, the trigger event has brought together multiple own-force awareness systems from different countries and from different force element types, including non-military forces. It is unlikely that any of these systems will accept inputs from the other systems without some changes to the systems or creation of some adaptor/gateway code, or a separate aggregation system In many cases, these systems will have difficulty representing any force element from another country or organization as an "own-force" element, if for no other reason than such forces are unlikely to have names or identifiers that fit into the own-force naming scheme. More likely they will look like "enemy" forces, or at best, "unknown" forces. In addition, a lot of the available force elements will probably have no "own-force awareness" system of their own. Attributes that characterize the size, equipment, and capabilities of force elements are also likely to be significantly different among whatever such systems might exist.

Policy issues might also make some participating systems reluctant to exchange information even if the transport and representation issues could be overcome. This is especially true for non-government organizations such as the Red Cross or Doctors Without Borders. Law enforcement organizations are also inhibited from sharing information with the military, civilian/commercial organizations, or across jurisdictional boundaries. The consequence is that there will be no single, cohesive picture of which disaster response force elements are where in the disaster area. Needless to say, this situation is rife with opportunities for misallocation of available resources to areas of need. Given this understanding, the truly surprising thing would be if such an event were handled smoothly by all the systems involved, even if they weren't impacted themselves by the physical destruction of the existing infrastructure.

We could explore this particular event type and the resulting system of system context in more depth, but the main lesson to learn from it is that events at the higher, broader levels of SoS scope are more likely to have significant interoperability impacts on the systems involved. These impacts are less susceptible to "quick fixes" or "patches," or to having humans intervene and bridge the gaps between system perspectives. The converse of this is that one may be able to predict the types of interoperability problems that may arise among some set of systems based on the type of trigger event considered. One can also anticipate what additional systems might be brought into contact with each other based on the type and scope of the trigger events considered. Therefore conducting an assessment of possible trigger events that might disrupt the scope assumptions of specific systems can allow us to engineer systems to operate more effectively in the SoS contexts they may find themselves in. Designing in flexibility and adaptivity to deal with possible trigger events may also create a more ecological mindset in system engineers rather than the more typical "draw hard/inviolate system boundaries" mindset. Whether this might actually make systems more capable of dealing with unanticipated trigger events (i.e., a broader range of such event types than were used in the assessments) is a topic that might be worth further research.

4.2 Oil Pipeline Failure

The second thought experiment considers a civil/commercial example which involves the events surrounding the 2 March 2006 oil pipeline failure in Alaska. A leak was discovered the Prudhoe Bay West Operating Area (PBWOA) pipeline operated by BP Exploration (Alaska) (BPXA) in a low-stress line. Approximately 5,000 barrels (210,000 gal) of processed crude oil spilled from a 0.25-inch by 0.5-inch hole in the failed pipe [5], resulting in the largest oil spill since the Exxon Valdez.

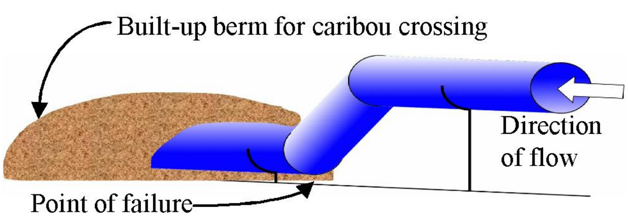

Figure 2. Diagram of PBWOA pipe at site of spill

Subsequent analysis and evaluation found a number of factors which contributed to the failure of the pipe [5]:

- High water/oil ratio
- Bacterial corrosion (increased corrosive agents in the crude oil)
- Exterior corrosion due to snow build-up
- Low velocity flow rate

While it is possible that any of these factors alone could have eventually led to failure, the important aspect is that their confluence precipitated this event. Here, there where (at least) three contexts which converged into this trigger event. The first context comes from the original design and production of the pipeline. The designers assumed certain characteristics regarding how much corrosion the pipeline would experience over its operational lifetime (technical/operational scope conflict). Next, the pipeline was operated under a different context, including 1) increasing the water/oil ratio to improve transfer efficiency and 2) operating at a low flow rate. This last factor has several interesting facets which merit further discussion. By lowering the flow rate, less oil could be transported (economic loss), but the physical stress on the pipe was relieved (technical gain). Further, the low rate allowed more water to separate out of the oil, carrying corrosive agents from the oil down to the bottom of the pipe (see Figure 2).

At this point, the third context enters the situation, that of the government oversight policies. Another ramification of operating the PBWOA line at a low flow rate is that it removed the line from federal regulation [6]. This meant that there was no federal oversight of the pipeline and, consequently, no mandated inspection requirements (policy). From the regulatory perspective, the line was "safe" since operating at high pressures results in greater stress on the pipe and increased danger of large spills from relatively small amounts of damage to the pipe (whether from corrosion or other source, such as gunfire). Another assumption of the regulatory policy is that the operator (BPXA) would conduct sufficient inspections in the absence of any federal requirements to do so (policy/operations scope conflict). In the case of the PBWOA line, the last time it was inspected was in 1998 and no established schedule was in place [5]. While inspection regulation may have reduced the possibility of failure, a separate regulation regarding the installation of the pipeline actually contributed to the failure (policy/technical conflict). While most of the line is held well above the ground, the location of failure was buried under a earth berm. This was due to a federal regulation regarding providing caribou crossings in the pipeline. As a result of this requirement, the pipeline was low enough to the ground that snow melt where the line enters the berm formed a puddle where exterior corrosion began.

This event shows the effects of conflicting perspectives across all of the scope categories. Table 1 provides some examples of entities evidenced in this trigger event and their location in the SoS framework. Note that, just as in the previous thought experiment, decisions and requirements at higher levels (such as the γ-level pipeline regulations) aggravated lower-level issues. Further, there was no system in place to identify this mismatch as operational decisions were made. While hindsight makes resolution of conflict easier, the point of the discussion—and even of SoS engineering—is to better foresee and diffuse possible trigger events to enable efficient interoperability. In this case, the multiplicity of perspectives pervaded the problem in such a way as to make each of the stakeholders believe that nothing was amiss when in fact a major incident was inevitable and unavoidable.

Table 1—Examples of pipeline terminology mapped to SoS lexicon

Level	Technical	Operations	Economics	Policy
α	Pipeline, well	Inspection program, oil extraction	Price of crude oil, extraction cost	Inspection requirements, caribou crossings
β	BPXA pipeline / well network	Pumping / routing / transferring oil	Local operating costs, refining costs	Integrity management programs
γ	Alaska pipeline network	Operators of Alaska-based oil extraction/refining	Price of refined oil, regional operating costs	Oil pipeline regulations
δ	National oil transportation network	National oil transportation operators	National demand / supply	All hazardous material transportation regulations

5 Conclusion

A perspective is a system's (or individual's) version of operational context. The implicit assumptions and scope boundaries that these perspectives carry pervade the individual systems, information models, and naming/identity conventions and authorities that institutional sponsors create. Combined with "trigger events" that bring multiple perspectives into contact, these assumptions can prevent achieving interoperability. A SoS is about interoperability among systems that have different perspectives and associated context and scope assumptions. Interoperability is complicated by context conflicts in complex situations usually precipitated by some trigger event.

Good SoS engineering will attempt to make perspectives and context/scope assumptions embedded in each system's interfaces and operations explicit, and, ideally, visible to the other systems it might interact with. It will also attempt to explore possible/probable trigger event types that might cause individual system perspectives and scope assumptions to come into conflict with each other, as well as introduce unanticipated systems into a given SoS context. Then the decision becomes whether the individual system should modify/expand its perspective and scope assumptions to deal with such events and new system types, or whether an adaptor/broker system approach should be used to isolate or buffer individual systems from the potential effects of such trigger event types.

Good SoS engineering will also realize that trigger events can impact SoS interactions at the individual entity representation level, at the overall conceptual business/institutional model level, or somewhere in between. In general, approaching SoS engineering with a good understanding of the power and influence of perspectives, context, and trigger events will result in more evolvable and open systems of systems that are more aware of their surroundings and context/scope assumptions. It will lead to more resilient and adaptive systems that respond more effectively to events that could have been anticipated—but were not—in engineering the individual systems.

Acknowledgements

The authors would like to express their thanks to the Pipeline and Hazardous Materials Safety Administration (PHMSA) and especially its administrator, VADM William Barrett for the inspiration for and insights into the pipeline discussion. That said, the discussion presented herein is the opinion of the authors and is in no way affiliated with or endorsed by PHMSA. We also wish to acknowledge the support of the NASA Graduate Student Researchers Program which provides a Fellowship for Don Fry (Grant NNL06AA00H).

References

[1] USAF Scientific Advisory Board, *Report on Systems-of-Systems Engineering for Air Force Capability Development*, SAB-05-04, July 2005.

[2] D. DeLaurentis and R.K. Callaway, "A system-of-systems perspective for public policy decision," *Review of Policy Research*, Vol. 21, No. 6, pp. 829–837, 2004.

[3] B. Nuseibeh, J. Kramer, A. Finkelstein, "A framework for expressing the relationships between multiple views in requirements specification," *IEEE Transactions on Software Engineering*, Vol. 20, No. 10, pp. 760–773, 1994.

[4] A. Finkelstein, *et al*. "Viewpoints: a framework for integrating multiple perspectives in system development," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 2 No. 1, pp. 31–58, 1992.

[5] S. Gerard, *Corrective Action Order* (CPF No. 5-2006-501H), U.S. Department of Transportations: Pipeline and Hazardous Materials Safety Administration, Office of Pipeline Safety, Washington, DC, 15 March 2006, accessed 8 March 2007 from http://ops.dot.gov/regions/west/BP%205-2006-5015H%20-%20Final.pdf

[6] T.J. Barrett, "Written statement to the oversight hearing on BP pipeline failure," Before the U.S. Senate Committee on Energy and Natural Resources, 12 September 2006, accessed 8 March 2007 from www.phmsa.dot.gov/news/SENATEEnergy-BarrettSept12.pdf