

Using a Boundary Object Framework to Analyze Interorganizational Collaboration

Allan Fong, Ricardo Valerdi, Jayakanth Srinivasan
Massachusetts Institute of Technology, Lean Aerospace Initiative
{afong05, jksrini, rvalerdi}@mit.edu

Copyright © 2007 by Allan Fong, Ricardo Valerdi, Jayakanth Srinivasan. Published and used by INCOSE with permission.

Abstract

The U.S. military is facing a plethora of challenges as a result of tightening procurement budgets and the need to acquire new capabilities to operate in modern war environments. This requires integrating legacy systems with developing technologies in what is loosely defined to be a System of Systems. Most Systems of Systems require some integrator to manage and operate the system interfaces. In addition to technical integration challenges, these system integrators have the difficult undertaking of integrating various organizations. The boundary object framework proposed by this paper provides a tool for systems integrators working in System of Systems or any type of complex system to identify and categorize communication, coordination, and collaboration interfaces and address possible failures.

Introduction

Technical and organizational interfaces are closely coupled factors for large scale programs involving multiple organizations. It is impossible to separate the two factors and understanding just one of these factors as it applies to a program is rarely enough to ensure the success of the program. As programs grow to involve more and more stakeholders, understanding the interactions between the human components of these systems become very relevant to the outcome of the program. In addition, programs and systems are evolving to become more complex and interrelated; this is especially true for the development of military systems in the US. It is becoming increasingly difficult to develop capabilities with a single system or with completely new technology. For example, to achieve their desired capabilities, the US military must integrate various existing war-fighting systems with the development of new systems into System of Systems (SoS) after which the desired capabilities can be realized. The U.S. Army's Future Combat System is an example of a SoS. The FCS is designed to have the capability of adjusting to a changing set of missions with flexible, agile, integrated, and sustainable systems, which will replace the legacy (heavy and slow) systems (Laird 2003). A SoS can reduce wasted resources and create value and capabilities that the components of the SoS cannot deliver individually. Although the collaboration amongst systems can result in greater value and capabilities, the integration and management of the SoS is much more difficult than managing the individual components and not very well understood.

While the majority of systems research has been focused at improving the technical portion, more work on the organizational dimension of system interactions needs to be done still. In addition to the technical models that already exist, frameworks to understand organizational interfaces in SoS are needed. This paper will address the issue of interorganizational interfaces

by applying the concept of a boundary object to previous research on organizational interactions and SoS. Understanding the interfaces in a SoS is critical because the greatest value from working in a SoS can be realized at the interfaces (Maier 1998). Studying the boundary objects used at the organizational interfaces is extremely important because these objects are the mediums through which information is shared between the different organizations. A plethora of researchers have studied the use of boundary objects within an organization, but these ideas have not been widely applied to the SoS context. This research will expand the boundary object theory to interorganizational interfaces involved in a SoS.

A boundary object framework is developed to define and categorize interorganizational interfaces based on a functional classification of the interface and a characterization of the boundary objects used at the interface. The framework is largely an integration of theories and results from past studies on organizational interactions, SoS, and boundary objects. This framework is then tested using a case study that looks at the Future Combat System. Relevant interfaces and boundary objects are defined and classified using this framework while providing an understanding into interface failures. The results from the case study are then used to improve the framework and customize it for SoS interfaces. An understanding of interorganizational interfaces will be extremely useful when architecting or rearchitecting a SoS.

System of Systems

Although there is a variety of definitions for System of Systems (SoS) (Lane and Valerdi 2005), most of these definitions agree that a SoS has to have components that are operationally independent and managerial independent (Maier 1998). Operational independence of a SoS component means that it must be able to independently operate in a useful manner if detached from the SoS. The managerial independence of the SoS component means that the component is actually operated independently even after it is separately acquired and integrated into the SoS. Maier explains these concepts in much greater detail in “Architecting principles for systems-of systems.”

The benefit and need of forming SoS have been realized and studied in many different applications. NASA’s space exploration initiative (Spurlock 2005), GEOSS (Lane and Valerdi 2005), a wide area network system, and an integrated air defense system (Maier 1998) can all be classified as SoS. In a SoS, new capabilities of the whole system can be realized that were not possible with each of the individual components working alone. Furthermore, SoSs are needed as technical systems become more complicated. This is especially true with the development of more complex and intricate military systems in the US (Moon 2005). Developing new capabilities is no longer solely about the development of the most advanced technology (Nicoll 2004); the value and capabilities resulting from SoS integration needs to be realized. This value greatly lies in the interfaces of the SoS components. However, these interfaces are also where the most danger exist (Maier 1998). As a result, the ability and execution in which the systems integrator manages these interfaces are critical to the success of the overall system.

Boundary Object Literature

Boundary objects are objects that are flexible enough to adapt to local needs and the constraints of the stakeholders employing them, yet specific enough to maintain a common identity across different interpretations. These objects have different meanings in different communities of practice, but their structures are common enough to more than one world to make them recognizable by a means of translation (Star and Griesemer 1989). Objects are

generally defined as the artifacts that a person or community works with (Carlile 2002; Greer et al 2006). These objects can be physical objects, such as architecture framework print outs, or electronic objects, such as e-mail. Another dimension of these objects is that they carry information, which can be explicit or implicit. For example, explicit information can be directly represented, such as on a blueprint or instruction manual, or information can be implied, such as the imbedded information in a product or picture. Boundary objects have been applied to many areas of research, i.e. from social science (Star and Griesemer 1989) to software development (John et al 2004; Gunaratne et al 2004). The boundary object concept have also been applied to CAD, sketches and drawings in the design engineering setting (Henderson 1991) as well as to project management tools to evaluate structures within an organization (Sapsed and Salter 2004).

Objects become boundary objects when they are used at the interface of different communities of practice. A community of practice has a shared understanding of what the community does, of how to do it, and of how it relates to other communities and their practices. A community of practice will develop the same world view or mental model (Brown and Duguid 1998). These communities of practices have been also referred to as social circles, stakeholders, organizations, etc. Boundary objects essentially exist and are used at the interfaces between these communities of practices. The figure below is a representation of the purpose of boundary objects. If designed and used properly, they can connected what was once separate communities together. The boundary object bridges allow the communities avenues to communicate, coordinate and collaborate. This paper categorizes organizations as communities of practices and focuses on the use of boundary objects at these community interfaces.

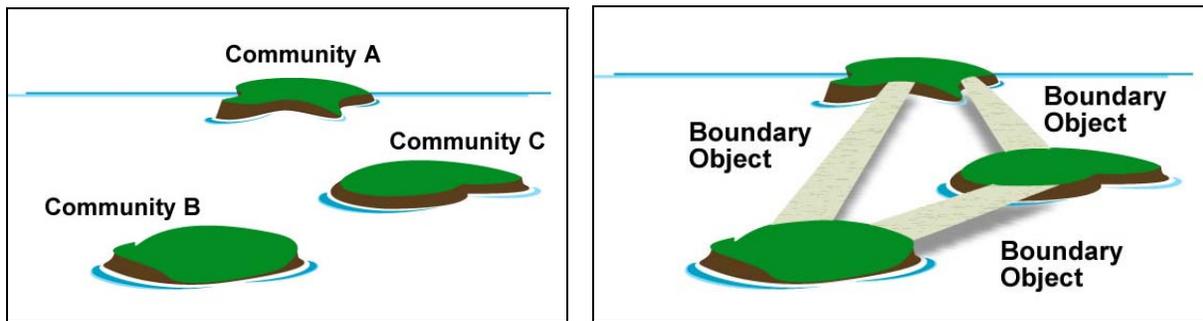


Figure 1a, 1b. Three communities independent from each other on the left and connected on the right

Furthermore, boundary objects carry information and context that can be used to translate, transfer and transform knowledge between communities of practice (Carlile 2004). The design and use of boundary objects are especially important when working between communities that are geographically separated, but should not replace face-to-face interactions. How well geographically distributed communities work together is correlated to how well they share information and knowledge at the interfaces (Sapsed and Salter 2004). Furthermore, these objects can also be dynamic. They can be changed and manipulated to carry more information or context. For example, a user can layer a boundary object, such as a requirements document, by highlighting certain phrases, writing comments in the margins, crossing out certain parts and so on (Swarts 2004). Each style of marking adds an additional layer to the object. As a result, the evolutionary characteristic of a boundary object and its ability to carry information and context, allow different communities interface (communicate, coordinate or collaborate) which

each other. The section below expands on how boundary object objects can be used to analyze communication, coordination, and collaboration at organizational interfaces.

Systems Integrators

The most value or leverage in constructing a SoS is at the interfaces (Maier 1989) and it is at these interfaces that the significance of boundary objects is realized. The value found at interfaces also extends to complex systems that are not necessarily SoSs. The value of a boundary object depends on how successful it can be used to decontextualize knowledge on one side of a boundary and recontextualize it on the other side. This process of decontextualizing and recontextualizing knowledge is dependent on the sharing and understanding of information during communication, coordination and collaboration interfaces; the following sections will describe these interfaces in more detail. The role of a systems integrator is, as the name implies, to integrate various systems together. Naturally, the systems integrator will care about how the boundary objects at these interfaces are used to integrate the information and knowledge amongst the different communities of practice.

In a SoS with no integrator, the different organizations can be thought of as unconnected islands. The Figures 2 and 3 further expand the previous Figure 1 of the bridged communities. Before the bridge boundary object is constructed, the different communities will have to work together or else they might end up with different bridge designs incapable of interfacing.

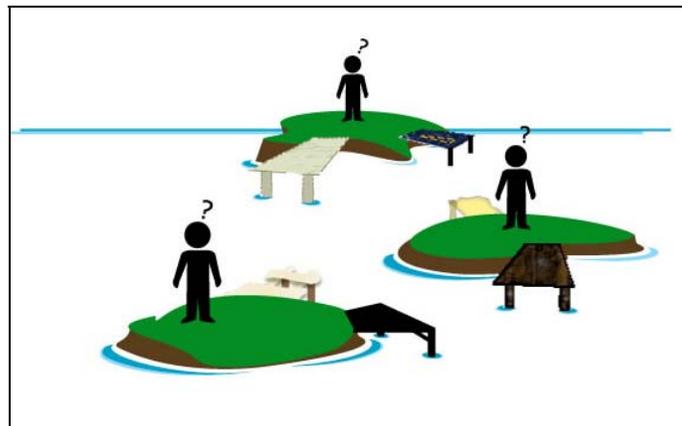


Figure 2. Collaborating is difficult without a systems integrator

The systems integrator needs to work within all the communities becoming the vital link that connects them. For example, the systems integrator has to make sure everyone is using the same units and compatible CAD software when designing and building their parts of the bridge. The bridge drawings and e-mails sent back and forth between the different communities are boundary objects use for collaboration purposes.

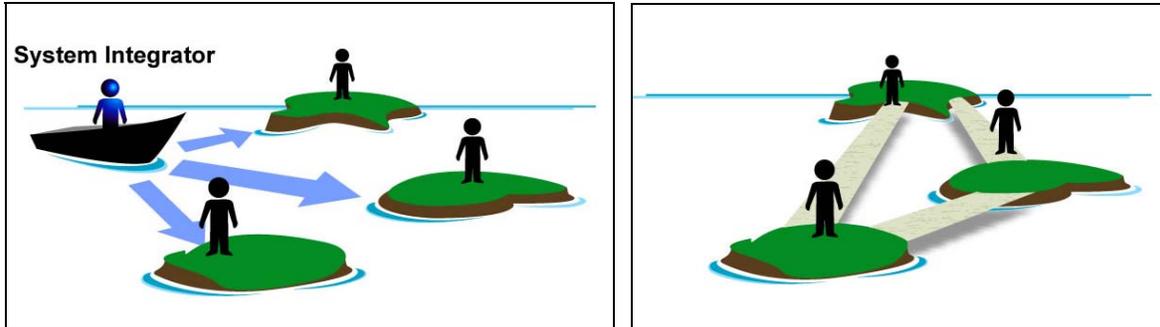


Figure 3a, 3b. System integrator works to connect the different communities which lead to more collaborative value and capabilities

By forming successful collaborative interfaces, the different communities will be able to design and build useful bridges. The bridges are another example of boundary objects that can allow more people, resources and information to flow between the communities resulting in more collaboration.

A systems integrator, or as it is sometimes referred to as, the lead systems integrator, needs to cultivate, develop, and maintain an environment in which the components of the system can develop, grow, and evolve. This includes providing a focal point for implementing proven best practices across the system and leveraging the work that is being done by other components in the system in a highly coordinated manner (Spurlock 2005; Gupta 2003). The system integrator must also develop boundary objects and maintain the environment in which these objects operate. In the previous bridge example, the system integrator has to make sure that the different communities can easily exchange information with each other when it is required. Furthermore, the integrator must create system awareness amongst the organizations by ensuring that boundary objects are used effectively for communication, coordination and collaboration purposes. Going back to the bridge example, before the initial construction of the bridge began, the system integrator must make sure all the communities can understand the information they receive from each other. For example, if each community spoke a different language, the system integrator must provide each community some method of translating each other's language. The systems integrator must be able to address failures in communication, coordination and collaboration between different organizations. The section below further elaborates on types of interfaces and their associated failures.

Communication

Communication is the process by which information is exchanged between individuals through a common system of symbols, signs, or behaviors (Merriam-Webster Online Dictionary). Similarly, a communication event has been defined as a verbal or non-verbal exchange between two or more people (Lingard et al 2004). In the context of SoS, an organization communication interface exists when two or more communities of practices transmit and receive information correctly. This can be a website used by the customer to post information about services it is seeking or it can be a prototype of a jet engine transferred between a supplier and contractor. Boundary objects are frequently used at these interfaces. The effectiveness of such interfaces depends on the participants, the location, the timing and content of the object. The participants refer to the question of who is at the interface and address the issue of whether or not the right people, communities, organizations are present. The location and timing factors of

communication refer to the physical and temporal situation of the interface (Lingard et al 2004). Lastly, content refers to the representation of information in the object used during the exchange and is divided technical and understanding concerns. For communication purposes, understanding depends on if the receiver has the necessarily skill-base to understand the information. It does not necessarily imply a common understanding between the sender and receiver. Table 1 below provides a categorization of the different types of communication failures associated with their respective elements.

Table 1. Summary of communication interfaces

Communication Interface			
Attribute	Question	Interface Failure Example	Boundary Object (in example)
Participants	Is the communication occurring between the right communities?	An office memo was accidentally sent to the contractors rather than the program manager	Office-memo
Timing	Is the communication occurring at the right time?	Requirements documents arrive late to the project manager	Requirements document
Location	Is the communication occurring in the right place?	Machine parts are delivered to the wrong storage facility	Machine parts
context (technical)	Does the information that was sent equal the information was received?	E-mail is corrupted after it is sent	E-mail
(understanding)	Did the receiver have the necessarily skill-set to understand the information?	Instruction manual is in Japanese rather than English	Instruction manual

Coordination

Coordination is the act or action of bring into a common action, movement, or condition (Merriam-Webster Online Dictionary). Coordination goes beyond communication by taking the information that was communicated and applying it for a purpose. All the conditions for communication apply to coordination interfaces. This definition of coordination is used in the SoS context to describe the interactions between different organizations rather than within just one organization. For organizations, coordination-type interfaces bring different communities of practices into common action or grounding, which describes both the physical and mental aspect of coordination. Coordinating common actions amongst organizations implies that there exists some kind of rule or condition that must be followed (Kogut and Zander 1996). For example, coordinating a conference means defining certain conditions such as when and where the conference will take place. The organizations involved in this event will have to abide by these conditions. In addition, coordination can also mean bring different communities into common grounding – shared understanding. In this paper, grounding refers to the interactive process by which organizations exchange evidence about what they do or do not understand in an effort to develop a common ground. The common ground shared by different communities is expanded through interactions. The participants in a successful coordination interface must identify what their partners are intending to do and monitor their partner’s level of comprehension (Fussell et

al 2000). This is different from just communication because communication did not require this comprehension feedback loop. Coordination needs not only a common syntax but a semantic approach that address different interpretations of language or context (Carlile 2002). At these interfaces, participants must be willing to change the knowledge and understanding from their own domain (Carlile 2004).

Coordination failures generally result from a lack of rules (Kogut and Zander 1996) and common grounding. Rules are important for coordinating procedures, such as flying. It would be very difficult to coordinate the thousands of airline flights each day without agreed upon rules and regulations. Furthermore, it is very difficult to have an effective coordination interface if there is a lack in common grounding between the participants. This failure can be attributed to the inability to identify the partner’s intent, comprehension of the situation, and poor feedback. Table 2 discussed different types of coordination interfaces and failures that can occur.

Table 2. Summary of coordination interfaces

Coordination Interface			
Attribute	Question	Interface Failure Example	Boundary Object (in example)
Rules	Are there agreed upon rules and regulations to follow?	Company was not able to attend conference because read the invitation incorrectly and missed the sign-up deadline	Conference invitation
Common grounding	Is there common understanding in the information being shared?	A program manager and engineer go to different conference rooms because the hand written note to the engineer did not clarify the location	Hand written note

Collaboration

Collaboration is the last of the functional interface categories. Collaboration is the process of working jointly with others or together especially for intellectual endeavors (Merriam-Webster Online Dictionary). Collaboration in the context of this paper applies to the interorganizational interactions within a SoS and has to have interorganizational trust, be beneficial to the partners, and lead to joint value creation (Zaheer et al 1998; Kanter 1994). It is important to note the distinction between interpersonal trust and interorganizational trust. Interpersonal trust is the trust that exists between an individual and his/her counterpart in the partner organization. Interorganizational trust, on the other hand, is defined as the extent of trust placed in the partner organization by another organization (Zaheer et al 1998). Furthermore, there are several types of trust that can exist between partners. The definition of trust that will be used in this paper will be (1) the confidence or predictability in one’s expectation and (2) the confidence in the other’s goodwill (Ring and Van de Ven 1992). Moreover, a collaborative interface must yield benefits for the partners and involve the creation of new value together (Kanter 1994). These criteria largely depend on the alignment of the partners’ goals, agendas, and motivations. A higher level of understanding between the partners is necessary to achieve effective collaboration. For example, it is not only necessary to understand requirement changes, but it is essential to understand why they were changed. In addition, the partners must want to work together and believe that such collaboration can offer new opportunities for them in the

future: this relationship is much more than just a simple deal (Kanter 1994). Furthermore, collaboration involves the creation of new value together rather than merely getting something back for what is put in (Kanter 1994). The realization of new value and new capabilities is a fundamental reason in developing System of Systems. Collaboration interfaces and failures examples are summarized in Table 3 below.

Table 3. Summary of collaboration interfaces

Collaboration Interface			
Attribute	Question	Interface Failure Example	Boundary Object (in example)
Interorganizational trust	Does the collaboration foster interorganizational trust?	Program manager does not trust explanation given by the supplier in their cost analysis	Supply cost list
Mutually yield benefit	Does the collaboration mutually benefit the partners and create new opportunities?	Interface between two organizations end after the termination of a temporary contract for sharing facilities	Contract
Value creation	Does the collaboration lead to the creation of new value together?	Contractor does not solicit feedback from end-users during their design review	Design review document
High level understanding	Do they understand the reasons behind decisions made their partners?	Sub-contractor receives a change in requirements from the prime contractor without any explanation	Changed requirements document

Boundary Object Framework

This paper proposes using a boundary object framework to identify and characterize organizational interfaces. It is important to note that although this framework has been originally conceived to analysis System of Systems, it can very easily be applied to complex systems, programs, projects, etc that require the collaboration between different communities of practice. This framework consists of two models that work together to understand and categorize these interfaces. The first model integrates previous studies on communication, coordination and collaboration into a single flowchart that can be applied to organizational interfaces. Once the interface is defined, the object or objects being used at the interface are identified and characterized by the second model, the boundary object model. This model takes a different approach to understanding organizational interfaces by giving a physical meaning to several variables necessary to understand organizational interactions. Variables such as understanding and trust become embodied in the objects used. Users of this framework can understand organizational interfaces more quantitatively. Furthermore, this framework will be a useful tool for systems integrators in understanding and diagnosing organizational interfaces failures

Interface Characterization Model. The Interface Characterization (IC) model determines the type of organizational communication, coordination, or collaboration interfaces that exist. Boundary objects play a key role in the IC model by providing flexibility and richness in the interactions. The model divides up interfaces into four functional types. Type A interfaces are

those in which bi-directional communication does not exist. The attributes of these interfaces are the participants, timing, location and context. Failure of this interface can be caused by unsuccessfully addressing one or more of these attributes. It is assumed that if communication is successful, all four attributes will be properly addressed. Type B interfaces are those that allow for communication but fail to provide means to coordinate between different communities. The coordination interface is associated with the implementations of rules and common grounding. Type C interfaces allow for communication and coordination but fail to sustain collaboration between partners. The attributes associated with the collaboration interfaces are interorganizational trust, mutually yielded benefit, value creation and a high-level of understanding. These attributes are also previously discussed.

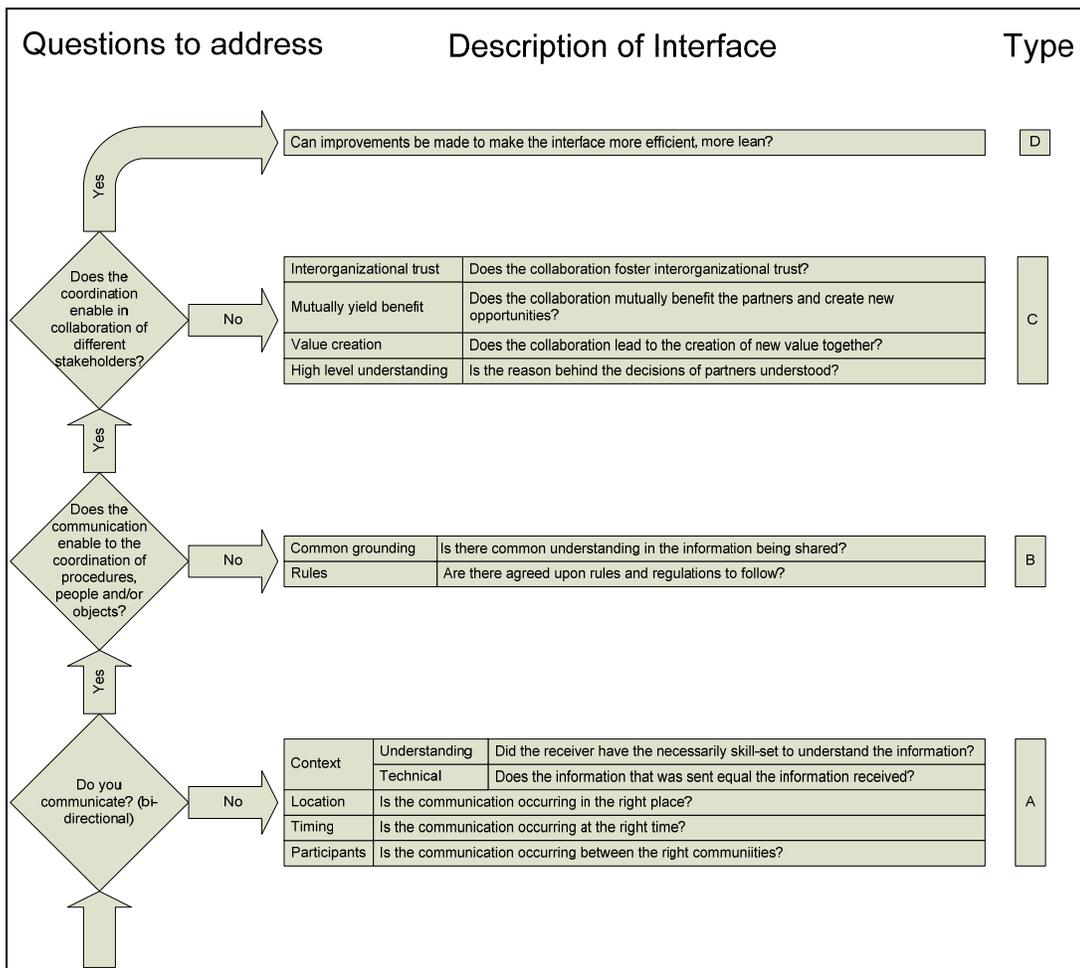


Figure 4. Interface Characterization Model

Organizations can determine what type of interface they want with other organizations. Sometimes interactions require a collaborative interface while other times perhaps only a coordination interface is required. For example, when the Air Force is soliciting proposals for a new contract, the program manager cares mostly about communicating their need out to industry. A collaborative partnership is not required at the time. However, once a contractor is awarded, the Air Force will begin to work closely with the contractor in developing the desired

technologies. In this case, the coordination interface has to evolve to include collaboration. The development and evolution of these interfaces can be represented by boundary objects, since these objects are the means in which information and knowledge is transferred.

In the context of a System of Systems, the role of the systems integrator is to manage these interfaces and ensure that the independently operated systems can successfully interface with each other. Choosing which boundary objects to use and how to use them become important tasks for the systems integrator. For example, organizations will not be able to share CAD models if they are using different modeling programs that are incapable of interfacing with each other. In this example, the systems integrator should set some rules about which programs to use if the organizations are to coordinate their models together. More attributes must be met for an effective collaborative interface to exist. These attributes can amount to cost of resources and time required to develop the boundary objects. Equally important is the systems integrator's ability to manage the creation and use of boundary objects that function at the interfaces. This issue will be addressed in the Boundary Object Characterization model.

Boundary Object Characterization Model. The Boundary Object Characterization (BOC) applies previous boundary object literature to characterize boundary objects based on ten attributes: medium, granularity, staleness factor, malleability, inclusivity, synchronization, importance, layers, context type, and traceability. The purpose of this model is to characterize the boundary objects used at specific interfaces. If a failure at the interface exists, the BOC model will be useful in categorizing the mode of failure and provide possible remedies. Furthermore, by considering the objects used at current interfaces, one can create new interfaces or modify existing ones to create more capabilities in the system.

Medium. Boundary objects will be distinguished by the medium in which they exist: virtual and physical. Virtual boundary objects are those that exist in bytes and bits. They are stored in computers, databases, palm pilots, etc and are transferred electronically. Examples of virtual boundary objects are e-mails, websites, and electronic databases. Physical boundary objects are objects that you can hold in your hands and physically manipulate, such as by highlighting, crossing-out and so on. Physical boundary objects are usually transferred from people to people such as being sent through the post office. Examples of physical boundary objects are hardcopies of design documents and aircraft clay prototype models.

Granularity. Granularity refers to the level of information detail and context represented in the object.

Staleness Factor. Staleness factor represents how stale, or not current, the information on the boundary object tends to be. It is generally defined by the average time to update the object divided by the average time between changes in the real-world information that is suppose to be represented in the object.

Malleability. This attribute describes the average level of effort and/or steps necessary for a user to modify the boundary object.

Inclusivity. Inclusivity describes how much input/output is used/given by the different stakeholders during the creation, use, and modification of the boundary object.

Synchronization. This attribute consists of two parts: external synchronization and internal synchronization. External synchronization describes the extent at which duplicates of the same object (i.e. document) are linked, such that a local change in one object will be propagated globally to all similar objects. Internal synchronization refers to the consistency of context within the object.

Importance. Importance simply refers to the criticality of the information represented in the boundary objects to the user. How and where the information would be applied are important factors for this attribute.

Layers. Layers refer to the additional tools and/or resources the user would need to understand the information in a boundary object (Swarts 2004).

Context Type. This attribute distinguishes between boundary objects used as open and closed tools. When an open object is used, onlookers can see how the user reached his/her conclusion “by making inferences based on the observable information contained in the tool [object]” (Swarts 2004). Conversely, a closed tool does not allow for such inferences.

Traceability. The traceability of an object is the level at which users can document and track alterations to the object. Traceability allows process transparency and accountability during the exchange and modification of boundary objects.

Implementation-Case Study

This framework is currently being used to analyze and categorize the organizational interactions in the Future Combat System (FCS). FCS has the characteristics of a System of Systems. In addition to being a collection of independent organizations, its role is to integrate existing legacy war fighting systems with new technologies. Surveys and interviews are currently being conducted amongst the different companies involved in FCS. A more detailed analysis of the FCS organizational interfaces will be provided later. The usefulness of the boundary object framework to analyze SoS will be addressed. Suggestions will be made on how to improve the framework and eliminate any latent problems in the two models.

Discussion

Initial interviews with representatives from various organizations part of FCS have shown strong correlations amongst trust, feedback, and face-to-face interactions at different types of organizational interfaces. In general, as an interface moves from communication to coordination to collaboration, the accountability, trust, and understanding between the participants increase. It has been found that boundary objects play critical roles in facilitating communication, coordination and collaboration. These objects are also used to decrease the amount of disconnects between different communities. Understanding the attributes of the boundary objects used has led to a better conception of the organizational interfaces.

Furthermore, it has been found that the boundary objects that are primarily used at organizational interfaces are produced using Microsoft Word, Excel and PowerPoint. These boundary objects exist both virtually, such as e-mail attachments or presentations, and physically, such as word documents and printed PowerPoint slides. It has also been observed

that the objects used at the interorganizational interfaces are not necessarily the same objects that are daily used by members of the same community. For example, an engineering company works primarily with SolidEdge to design and model a component of a vehicle. However, when presenting to the Army, the company will generally place photos of the modeled vehicle component in a PowerPoint presentation without using SolidEdge to present the model. Moreover, trust and understanding decisions amongst FCS members are issues that have been raised. This research aims at unraveling these issues through further interviews and study.

This framework also illustrates that changes in organizational interfaces are marked with changes in the boundary objects used at the interfaces. Figure 5 below shows one example of how the usage of boundary objects change with the purpose of the interface. Further results and discussion will depend on additional interviews and data.

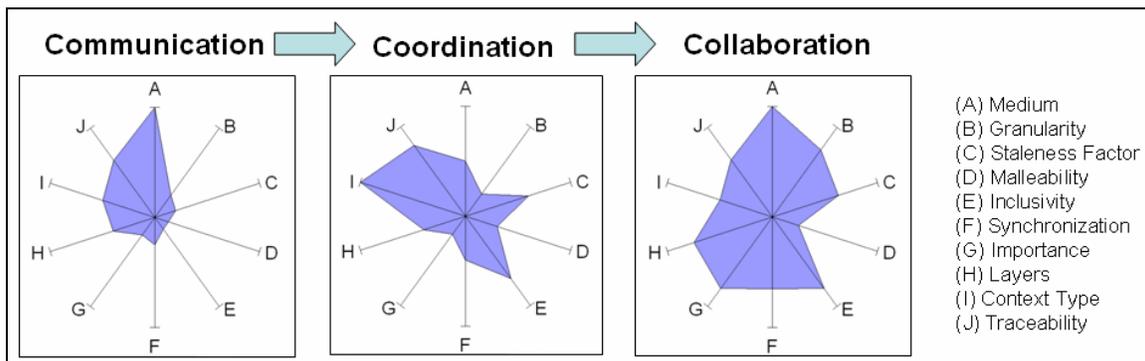


Figure 5. Example of how evolving interfaces influence the use of boundary objects

A limitation of this framework is the inability to capture verbal interfaces. There are times when people meet and talk without any documentation or record. The information that gets exchanged is stored in the people's memories and unless it is contextualized in some way, it has the high possibility of getting lost either proactively or retroactively. Boundary objects have been viewed as memory markers for organizations (Ackerman and Halverson 1999) and can help in performance of organizational interfaces. If no object is used, the interface can still be characterized by the IC model but the BOC model will no longer be applicable. Furthermore, applying this framework requires a good understanding of the environment in which the SoS operates, requiring resources and time. This is, however, very important in understanding the nature of different organizational interfaces. Nevertheless, it is believed that this framework can be used as a building block to understand complex networks of organizational interfaces. Refinement of the framework is needed and it needs to be applied to more case studies. Hopefully with further use and useful feedback, the framework can be improved to capture organization interface behaviors more accurately.

Conclusion

Organizational interfaces often parallel technical interfaces. This research offers a unique way to model and categorize organizational interfaces by using a boundary object framework. This framework consists of two models, one to categorize organizational interfaces and the other to characterize the boundary objects used at these interfaces. Using boundary objects correctly is essential to any complex system, program, project, etc, that involves the collaboration of multiple communities of practice. By understanding the attributes of these objects, one can

obtain a better comprehension of the boundary interface and address potential disconnects that might exist. The case study has shown that this framework can be applied to model organizational interfaces and the objects used for communication, coordination, and collaboration. This framework can be extended for systems integrators managing the organizational interfaces in most any complex system.

Biographies

Allan Fong is a graduate student at Massachusetts Institute of Technology in the Department of Aeronautics and Astronautics engineering. His research and his studies are funded by Lean Aerospace Initiative. He will be completing a Masters of Science by May 2007 and is intending to pursue a doctoral degree. Allan graduated from Columbia University in 2005 with a bachelor's degree in mechanical engineering.

Ricardo Valerdi is a Research Associate with the Lean Aerospace Initiative at MIT. He is currently the research lead for the Enterprise Cost and Metrics cluster. Ricardo received his doctoral degree in systems engineering from USC in 2005, where he created for the COSYSMO model for systems engineering cost estimation.

Jayakanth "JK" Srinivasan is a Research Associate with the Lean Aerospace Initiative at MIT. He is currently the research lead for the Enterprise Integration enabled by IT cluster, focusing on IT Architectures and Lean Software Development. JK has an undergraduate degree in Computer Engineering and graduate degrees in Avionics and Aeronautics and Astronautics respectively.

Acknowledgements

This research would not have been possible without the support of the LAI consortium members and The Aerospace Corporation. The authors of this paper would also like to acknowledge the support of Richard Adams, Bonnie Troupe and Diana Fong. Furthermore, acknowledgements and thanks are extended to the individuals who participated in this study.

References

- Ackerman, Mark S., and Halverson, Christine, "Organizational Memory: Processes, Boundary Objects, and Trajectories," 32nd Hawaii International Conference on System Sciences Proceedings, 1999.
- Bar-Yam, Y., and Kuras, Michael L. "Complex Systems and Evolutionary Engineering." Concept paper The MITRE Corporation, 2003.
- Brown, John S. and Duguid, Paul. "Organizing Knowledge." California Management Review Vol. 40 (1998): 28-44.
- Carlile, Paul R. "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development." Organization Science Vol. 13 (2002): 442-455.

- Carlile, Paul R. "Transferring, Translating, and Transforming: An Integrative Framework for Managing Knowledge Across Boundaries." Organization Science Vol. 15 (2004): 555-568.
- Fussell, Susan R., Kraut, Robert E., and Siegel, Jane. "Coordination of Communication: Effects of Shared Visual Context on Collaborative Work." CSCW'00 December 2000, PA.
- Greer, Donald R., Adams, Richard J., and Black, Laura J. "Improving Inter-Organizational Baseline Alignment in Large Space System Development Programs." IEEEAC December 2005, 2006.
- Gunaratne, Junius., Hwong, Beatrice., Nelson, Christopher., and Rudorfer, Arnold. "Using Evolutionary Prototypes to Formalize Product Requirements." Siemens Corporate Research. NJ, 2004.
- Gupta, Amar. "Role and Importance of Lead System Integrator in Context of New Air Operations Centers." White paper MIT Sloan School of Management, 2003.
- Henderson, Kathryn. "Flexible Sketches and Inflexible Data Bases: Visual Communication, Conscriptioin Devices, and Boundary Objects in Design Engineering." Science, Technology, and Human Value Vol. 16 (1991): 448-473.
- ISO/IEC, "International Standards." First Edition 2002.
- John, Bonnie E., Bass, Len., Kazman, Rick., Chen, Eugene. "Identifying Gaps between HCI, Software Engineering, and Design, and Boundary Objects to Bridge Them." Workshop CHI 2004. Vienna, 2004.
- Kanter, Rosabeth M. "Collaborative Advantage." Harvard Business Review July-August 1994: 98-108.
- Kogut, Bruce and Zander, Udo. "What Firms Do? Coordination, Identity, and Learning." Organization Science Vol. 7 (1996): 502-518.
- Laird, Robbin F. "Transformation and the Defense Industrial Base: A New Model." Defense Horizons May 2003. 16 Dec. 2005.
<http://www.ndu.edu/inss/DefHor/DH26/DH26.htm>
- Lane, Jo Ann and Valerdi, Ricardo. "Synthesizing SoS Concepts for Use in Cost Estimation." International Conference on Systems, Man and Cybernetics Proceedings, 2005.
- Lingard, L., Espin, S., Whyte, S., Regehr, G., Baker, G. R., Reznick, R., Bohnen, J., Orser, B., Doran, D., Grober, E. "Communication failures in the operating room: an observational classification of recurrent types of effects." Qual. Saf. Health Care 2004 Vol. 13 (2004): 330-334.

Maier, Mark W. "Architecting Principles for Systems-of-Systems." The Aerospace Corporation CH 1-460: 1998.

Merriam-Webster Online Dictionary, October 2006. <http://www.m-w.com>

Moon, David and Schoder, Alexandre. "Give Companies Incentive to Seek LSI Role." Defense News.com 2 May 2005. 18 Nov. 2005.
<http://www.defensenews.com/story.php?F=821171&C=commentary>

Nicoll, Alexander. "The Changing Role of Defence Firms." India Limited. 2003. 18 Nov. 2005. <http://www.rediff.com//money/2004/oct/08guest1.htm>

Ring, Peter S. and Van de Ven, Andrew H. "Structuring Cooperative Relationships between Organizations." Strategic Management Journal Vol. 13 (1992): 483-498.

Sapsed, Jonathan, and Salter, Ammon. "Postcards from the Edge: Local Communities, Global Programs and Boundary Objects." Organization Studies Vol. 25 (2004): 1515-1534.

Spurlock, Darren M. "Space Exploration Systems Integration." 1st Space Exploration Conference: Continuing the Voyage of Discovery Proceedings, 2005.

Star, Susan L., and Griesemer, James R. "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39." Social Studies of Science Vol. 19 (1989): 387-420.

Swarts, Jason. "Textual Grounding: How People Turn Texts into Tools." J. Technical Writing and Communication Vol. 34 (2004): 67-89.

Zahneer, Akbar., McEvily, Bill and Perrone, Vincenzo. "Does Trust Matter? Exploring the Effects of Interorganizational and Interpersonal Trust on Performance." Organization Science Vol. 9 (1998): 141-159.