Systems Research Forum

School of Systems and Enterprises
Stevens Institute of Technology

Printed in the United States of America.

Library of Congress Cataloging-in-Publication Data

ISBN-10: 0-9787122-1-8

Published by:
SSE Press
School of Systems and Enterprises
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030
www.stevens.edu/sse

Contents

Forum Focus and Editorial Board

The Systems Research Forum is dedicated to providing a platform for peer-reviewed graduate and post-graduate research papers and case studies in systems engineering. We invite original research papers addressing the various aspects of systems engineering and architecting, system analysis and evaluation, enterprise architecting and management, measurement and metrics, and simulation and modeling.

All papers will be peer reviewed, and by submitting a manuscript, the author certifies that it has not been copyrighted or previously published and that it is not currently under review for another publication.

Author guidelines are contained at the back of this issue, any questions regarding the Systems Research Forum can be sent to one of the Editors-in Chief:

Dr. Rashmi Jain
Stevens Institute of Technology
School of Systems and Enterprises
Castle Point on Hudson
Hoboken, NJ 07030
Tel: 201.216.8047
rashmi.jain@stevens.edu

Dr. Brian Sauser
Stevens Institute of Technology
School of Systems and Enterprises
Castle Point on Hudson
Hoboken, NJ 07030
Tel: 201.216.8589
brian.sauser@stevens.edu

Modeling, Analysis and Implementation of Infrastructure for Model-Based Integration and Testing

N.C.W.M. Braspenning, J.M. van de Mortel-Fronczak, and J.E. Rooda
Eindhoven University of Technology, The Netherlands

Abstract

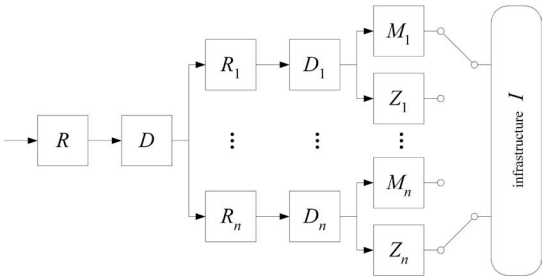
To reduce the lead time and the costs of integrating and testing high-tech multi-disciplinary systems, we use formal and executable models of components for early system analysis and for early integration and testing with available component realizations. In this paper, we investigate the role of the infrastructure that establishes the interaction between the components. We include a model of the infrastructure for early system analysis and we implement a corresponding integration infrastructure to integrate and test models and realizations. Application of this approach to examples of typical interaction types proves to be rather straightforward, allowing proper analysis of system and infrastructure properties, which remain valid during model-based integration and system testing.

Introduction

To reduce the ever increasing lead time and costs of integration and testing in high-tech multi-disciplinary system development, we propose a *model-based integration and testing* (MBI&T) method as described in (Braspenning et al. 2008). The method is illustrated in Figure 1, showing the system development process that starts with requirements R and design D of the system. Subsequently, requirements R_i, designs D_i, models M_i, and realizations Z_i of all n components of the system are developed. The components, represented by either a model M_i or a realization Z_i (depicted by the 'switches'), should interact according to system design D in order to fulfill the system requirements R. The component interaction as designed in D is realized by integrating components via an infrastructure I, e.g., using nuts and bolts (mechanical infrastructure), signal cables (electronic infrastructure), or communication networks (software or model infrastructure).

To detect and prevent integration problems at an early and therefore less expensive (Boehm and Basili 2001) stage of the development process, several model-based analysis and testing techniques can be applied in the MBI&T method. For example, simulation and model checking can be used to validate and verify the behavior of the system model (i.e., with models only), and testing can be used to find problems in a partly realized system (i.e., with models and realizations).

Figure 1. MBI&T Method



In our project, we model the components in a process algebraic language (Baeten and Weijland 1990). The behavior of a process algebraic model is fully specified by formal semantics. This enables proving the correctness of a model, e.g. model checking of deadlock, livelock, safety, and other behavioral properties. Communication in the process algebraic language is synchronous, i.e., corresponding send and receive actions take place simultaneously. Using synchronous communication reduces the complexity of the model, resulting in a better understanding of the system behavior. Furthermore, it reduces the number of states in the model which improves the capabilities of model checking.

In this paper, we investigate the modeling, analysis, and implementation of component interaction via infrastructure I. Although the models use synchronous communication, real systems often use asynchronous communication, i.e. send and receive actions do not take place simultaneously. This means that the analysis results based on models using synchronous communication, e.g., correctness of behavioral properties derived from the system requirements, do not necessarily remain valid when the models are used for integration and testing with realizations in an asynchronous environment.

Literature provides several approaches that deal with correct implementation of synchronous models in an asynchronous environment. However, these approaches cannot be applied in the MBI&T method since the perspective on the goal of modeling is different. In

most approaches found in literature, the models serve as basis for the realizations (software only), and they need some adaptations before they can be implemented in an asynchronous environment. For example, some approaches require that the model specifications are restricted to a certain modeling language subset (Mörk 2001) or a protocol should be added to negotiate which components will communicate (Demaine 1998). A common challenge in these approaches is the correct implementation of the non-deterministic choice operator (Palamidessi 1997), since this may offer many communication alternatives of which only one may be selected.

In contrast to these approaches, the MBI&T method focuses on finding problems in the system *as it is designed* by the engineers. This means that the models are based on the "as is" designs of the components (both hardware and software) and infrastructure. When the above mentioned approaches would be applied, the models would need to be adapted for asynchronous implementation, e.g., language constructs outside the implementable subset would have to be removed or some protocol for communication negotiation would have to be added. This means that the models would deviate from the "as is" designs, which does not suit the MBI&T method. Using the "as is" designs as basis for modeling also means that when a non-deterministic choice appears in a component design, it must also be modeled "as is" such that potential problems caused by it in an asynchronous environment can be analyzed. In our view, solving these problems is not part of the modeling (as in the approaches found in literature), but of the design activities. Of course, the approaches found in literature can still be applied to the design (and subsequently to the model) in order to solve the problems.

In the MBI&T method, the asynchronous component interaction as designed in system design *D* and realized in infrastructure *I* is expressed in the synchronous modeling language, which is a long known approach (Milner 1989). In this way, we can use the powerful techniques available for synchronous models to analyze the behavior in an asynchronous environment. To integrate and test models and realizations, an asynchronous infrastructure is used that implements the communication behavior as it was designed and modeled. This is done in such a way that the analysis results of the synchronous system model remain valid when integrating and testing models and realizations in an asynchronous environment.

The structure of the paper is as follows. The next section contains an overview of the forms of infrastructure *I* used in the MBI&T method. Subsequently, practical examples of modeling, analysis, and implementation of typical interaction types are given. The last section contains some concluding remarks.

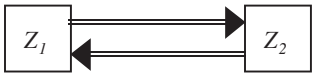
Infrastructure in the MBI&T Method

The MBI&T method consists of three main activities: modeling the components and their interaction, analysis of the resulting system model, and testing of integrated models and realizations of components. In these activities, the infrastructure is used in three different forms: infrastructure realization, infrastructure model, and model-based integration infrastructure. This section describes these different instantiations of the generic infrastructure *I* depicted in Figure 1.

Infrastructure Realization I_Z.

This is the "real" infrastructure that implements the component interaction according to system design *D*, e.g., via cables and communication networks. The example in Figure 2 shows two component realizations Z_1 and Z_2 (boxes) and the infrastructure realization I_Z (double lined arrows) that enables the communication between the components. Because I_Z is part of the real system in the real world, communication is asynchronous.

Figure 2. Infrastructure Realization I_z



Infrastructure Model I_M

In the MBI&T method, the system components are modeled as M_i (see Figure 1). Besides the components, also the infrastructure that enables component interaction is modeled and analyzed.

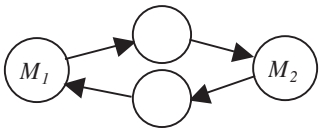
The infrastructure can be modeled on different abstraction levels. During the initial modeling and analysis phases, there may be reasons to use synchronous communication in the model, i.e., completely ignoring the asynchronous behavior. One reason may be that a detailed infrastructure design is unavailable, but system model analysis with a synchronous abstraction of the infrastructure is still helpful. Another reason may be that the infrastructure details are not important for certain model-based analysis activities and would only increase the complexity and state space when they are included in the model. For example, analyzing the functionality of the system may be possible when only the result of an interaction is known (e.g., a message being transferred), without knowing exactly how that interaction is established.

Although the asynchronous infrastructure behavior may be ignored in the model initially as described above, it must be considered eventually. After all, the models developed in the MBI&T method will eventually be integrated and tested with realizations that do require an asynchronous infrastructure. It is important to ensure that

the behavioral properties of the analyzed system model are still valid when the component models are integrated and tested with component realizations. Suppose that a system model with a synchronous abstraction of the infrastructure is found to be correct during analysis. Subsequently, some component models are replaced by the corresponding realizations, which require an asynchronous infrastructure. The resulting model-based integrated system is then used for testing. Due to the different infrastructural behavior, the models might also interact differently with the other components, possibly resulting in wrong conclusions about the test results. Even worse, when certain safety requirements checked during model-based analysis are influenced by the infrastructure behavior, safety is not guaranteed in the realization environment, possibly resulting in hazardous situations.

When the infrastructure details are taken into account during the modeling and analysis of the system, the asynchronous behavior of the real infrastructure I_Z must be expressed in the modeling language that is used. For certain interaction types, the modeling language may have constructs to directly express that type of infrastructure. For interaction types that cannot directly be expressed in a modeling language, it may be possible to model their equivalent behavior. In the case of the process algebraic language used in the MBI&T method, the asynchronous communication can for instance be modeled in the synchronous modeling language (Milner 1989). Additional processes are placed between two component processes to model the behavior of that particular component interaction. Different types of component interaction may require different additional processes in the model, as shown for some examples in the next section of this paper. We denote the modeling constructs used to express the component interaction behavior as the infrastructure model I_M. The example in Figure 3 shows four processes (circles) which, conforming to the process algebraic language, use synchronous communication (single lined arrows). The processes M_1 and M_2 represent the component models. The processes between M_1 and M_2 represent the infrastructure model I_M, resulting in asynchronous communication behavior between component models M_1 and M_2.

Figure 3. Infrastructure Model I_M



Model-Based Integration Infrastructure I_{MZ}

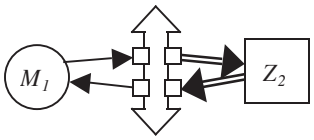
Besides the infrastructure realization I_Z and the infrastructure model I_M, another form of infrastructure is needed in the MBI&T method. A so-called model-based integration infrastructure I_{MZ} is used to integrate combinations of models and realizations, i.e., implementing the component interaction as designed in D and modeled in I_M. To enable this, I_{MZ} should satisfy several requirements.

First of all, the communication paradigm of I_{MZ} should be asynchronous, since the realizations which are integrated by it will also communicate asynchronously. Furthermore, different types of component interaction may require different behavior from the infrastructure. Therefore, these different interaction types should be supported by I_{MZ}, similar to the different behavior that can be modeled in the infrastructure model I_M. Finally, the model-based integration infrastructure should allow easy integration of models and realizations. This requires that both models and realizations can be connected to the infrastructure with minimal effort. To achieve this, the connection of components to the infrastructure should be independent of the form (model or realization) of the other components and of their exact name, location and interfaces. This makes the integration of components independent of whether models or realizations are used.

The last requirement, independency of connected components, is one of the main features of so-called *middleware*, which consists of intermediate software that connects software components with each other. The components only need to connect and communicate with the middleware and do not depend on the form, name, and location of the other components. In the MBI&T method, the model-based integration infrastructure I_{MZ} is also based on middleware. An example is given in the next section of this paper.

Connecting components to middleware requires that the communication paradigms used by the component models or realizations are adapted to the communication paradigm of the middleware. This is done by creating "connectors" for the models and realizations such that they communicate via the communication paradigm of the middleware. Different types of components, e.g. software components developed in different languages and tools or hardware components, may require different connectors to be created. We denote the middleware together with the connectors for the models and realizations as model-based integration infrastructure I_{MZ}. The example in Figure 4 shows the integration of a model M_1 and a realization Z_2 using middleware (vertical double headed arrow). Both components are connected to the middleware via connectors (small boxes) that adapt the communication paradigm of M_1 (single lined arrows) and the communication paradigm of Z_2 (double lined arrows) to the middleware. The middleware is configured such that the component interaction corresponds to that of Figures 2 and 3.

Figure 4. Model-Based Integration Infrastructure I_{MZ}



With these three forms of infrastructure, the MBI&T method can be summarized in the following procedure. This procedure takes the component designs D_i and the infrastructure design (usually part of system design D) as a starting point and consists of three phases.

1. Modeling
   a. Components M_i based on D_i.
   b. Infrastructure I_M based on D, if available and important for the analysis.
2. Model-based system analysis
   a. With synchronous abstraction of the infrastructure.
   b. With infrastructure model I_M.
3. For each realized component Z_i:
   a. Replacement of model M_i by realization Z_i, using model-based integration infrastructure I_{MZ}.
   b. System testing of the integrated system obtained in 3a.

In the following section, we practically illustrate the modeling, analysis and implementation of interaction types typically used in the high-tech multidisciplinary systems we are considering in our project, taking the wafer scanner from ASML (ASML 2008) as an industrial example.

Examples

In our project, we mainly focus on the interaction and time behavior of concurrent processes. This behavior is an important aspect of software and electronic components and strongly relates to the interaction between these components. In general, concurrent behavior is less relevant for mechanical components, and these components themselves are often controlled via electronics and software. Therefore, we concentrate on software and electronic components and their interaction in the modeling and analysis of concurrent behavior.

In the following paragraphs, we give examples of the main software and electronic interaction types used in the ASML wafer scanner, namely function calls in software and sequential logic in electronics. For each interaction type, we explain the behavior and properties of the infrastructure realization I_Z and show how this behavior can be captured in a synchronous process algebra model I_M. The system model with all component models M_i and the infrastructure model I_M is used to analyze behavioral properties of the system and the infrastructure. Subsequently, we show how each interaction type can be implemented in the model-based integration infrastructure I_{MZ} using middleware.

In the examples, we use the process algebraic language χ (Chi) (van Beek et al. 2006) and its toolset (Systems Engineering Group 2008) to model and analyze components and systems. As demonstrated in an industrial case study (Braspenning et al. 2008), the χ toolset allows simulation and model checking of system models, as well as real-time execution of component models integrated with other (non-χ) components via middleware (Millard et al. 2006).

The middleware used as basis for the model-based integration infrastructure I_{MZ} in the MBI&T method and in the χ toolset is based on communication via the publish-subscribe paradigm (Eugster et al. 2003), which satisfies all requirements for I_{MZ} defined in the previous section. The publish-subscribe paradigm is suitable to decouple the components, because the components do not need to know the exact form, name, location, and interfaces of the other components. Communication via the publish-subscribe paradigm is simple. Components can publish messages of a certain type (also called topic) to the middleware, and they can subscribe to message types published by other components. Communication via a publish-subscribe middleware is asynchronous since a message is first published to the middleware by a sending component, and then delivered by the middleware to the subscribed components. Different types of component interaction, also modeled in different models I_M, can be configured by quality of service (QoS) properties like the number of messages to keep as history, or the reliability of message delivery. Finally, both models and realizations can easily be connected to the publish-subscribe middleware. The connectors for a model of a component must relate all send and receive actions of the model to the corresponding write actions (for published message types) and read actions (for subscribed message types) of the publish-subscribe middleware. The χ toolset includes an automatic generator of connectors for a χ model of a component. The connectors for a component realization depend on the components themselves and may for example involve adapters that translate subscribed messages to function calls and function replies back to published messages, or software-hardware adapters that translate between software messages and electronic signals.

Function Calls (Software)

A wafer scanner is controlled by a large amount of software, consisting of more than 12 million lines of code. The main interaction type used in this software system is the function call. A function call consists of an asynchronous request from a client to a server that provides the requested function, followed by waiting for an asynchronous reply

from the server with the results of the function. The "wait for reply" action can possibly contain a time-out that is triggered when the reply is not received within a specified amount of time. In practice, these time-outs are used to detect errors in the function execution by the server.

There are two different types of function calls, blocking and non-blocking. In a blocking function call, no other statements may be executed between the request and the reply, while this is allowed in a non-blocking function call. Since the blocking function call is a special case of non-blocking (with no statements between request and reply and no time-out), we will only discuss the more generic non-blocking function call here.

Important properties of function calls are:

- FIFO order: For requests and replies between client and server and vice versa
- Buffer size: Limited number of messages in asynchronous communication buffer
- Consistency: the number of requests is equal to the number of replies or at most one larger (during function execution)
- Wait/time-out: A time-out may only be triggered when the reply buffer is empty for the specified amount of time since the start of the "wait for reply" action.

Note that using the time-out as an error detection mechanism could be captured in a property "time-outs may never be triggered," however this property does not relate to infrastructure but to required system behavior.

Function calls can easily be modeled in χ. Asynchronous communication can be modeled in a synchronous modeling language such as χ by including a "buffer" process between two communicating processes. The χ code of a buffer process B is shown in Figure 5. This process has two communication channels, input *a* and output *b*, for messages of type *msg*, and shows repetitive behavior (denoted by *). Each repetition starts with guarded expressions (denoted by ->) to check for buffer overflow, i.e., whether the length of message list *xs* exceeds the configured buffer size *n*. If this is not the case, the process continues its behavior (denoted by *skip*). The buffer overflow check is followed (denoted by sequential composition ;) by two alternatives (denoted by |) of which the one that is enabled first will be selected. Either a new message *x* is received via channel *a*, which is then appended to *xs*, or, if *xs* is not empty, the head (first item) of *xs* is sent via channel *b*, after which the tail (all but first item) of *xs* remains.

Using multiple instantiations of buffer process B, we can model a function call as shown in Figure 6. The χ code shows four processes in parallel composition (denoted by ||). The first process is a partial specification (denoted by …) of a client that calls some function *f*. This function call is modeled as a sequential composition of sending an asynchronous request with the function arguments

Figure 5. Buffer Process B

```
proc B(chan a?, b!: msg, val n: nat) =
|[ var xs: [msg] = [], x: msg
:: *( ( len(xs)  > n -> !!"buffer overflow"
      | len(xs) <= n -> skip
      )
    ; ( a?x; xs:= xs ++ [x]
      | len(xs) > 0 -> b!hd(xs); xs:= tl(xs)
      )
    )
]|
```

(*f_req!arg*) and receiving an asynchronous reply of the function with the results (*f_rep?res*). Between these two statements, other internal actions (denoted by …) may be performed (not for blocking function calls). The possible time-out on the 'wait for reply' action is modeled as an alternative composition of the receive action and a delay of *t* time units, which means that either the reply is received or the delay is finished, resulting in a time-out. Note that *t* is infinity (no time-out) for blocking function calls. The second process models the server, which repetitively waits for requests for the only function it provides, function *f* (more provided functions can be added in a similar way). Upon receiving a function call request from a client with certain arguments *arg*, the result of the function executed on *arg* is sent back as a reply. Finally, two buffer processes B are used to model the asynchronous communication. The buffer processes are connected to the request and reply channels of the client and server, similar to Figure 3. The buffer sizes are set to one since a client process may only call one function at a time. To simplify the example, we assume that a function is required by only one client. A function required by multiple clients can be modeled in a similar way.

Figure 6. I_M for Function Call

```
(  ( ...
   ; f_req!arg
   ; ...
   ; (f_rep?res | delay t -> !!"time-out")
   ; ...
   )
|| *( f_req'?arg; f_rep'!f(arg) )
|| B(f_req,f_req',1)
|| B(f_rep',f_rep,1)
)
```

Using this infrastructure model I_M for function calls, we can include the infrastructural properties mentioned earlier in this section during system model analysis.

Due to the use of lists and their head and tail functionality in the buffer processes, it is not possible for two messages to overtake each other in the buffer, so FIFO behavior is guaranteed. The validity of the buffer size property depends on the behavior of all components, and can be checked by performing a reachability analysis of the buffer overflow state of all buffer processes, e.g. by using a model checker as in (Braspenning et al. 2008). In the

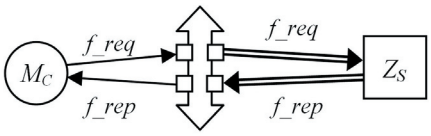
model of the server, each incoming request is immediately followed by sending the reply, so the number of requests is always equal or at most one larger than the number of replies. For more complex server models, e.g. with functions required by multiple clients, request and reply counters can be added to enable model checking of the consistency property: $0 \leq$ nr_requests – nr_replies ≤ 1. The wait/time-out property is covered in the infrastructure model I_M of Figure 6, because the communication in the χ model is urgent, i.e. a process may not delay if a communication action is enabled. This implies that the time-out (*delay t*) can only be triggered when after *t* time units the receive action (*f_rep?res*) has not been enabled. Besides these already listed properties, two properties of blocking function calls, namely subsequent requests and replies (no intermediate statements) and infinite time-outs, can be checked by static analysis of the model structure (e.g., by a compiler).

When integrating models and realizations of components that use function calls to interact, the model-based integration infrastructure I_{MZ} can easily be implemented in the publish-subscribe middleware. Since the middleware uses asynchronous communication itself, it is well suited as implementation of the buffer processes B from I_M and of the real buffers used in the real function calls in I_Z. Figure 7 shows the implementation of I_{MZ} for the example of Figure 6 with a model of client M_C, a realization of server Z_S, and the required message types for requests and replies. The client is configured as publisher of requests for function *f*; and it is subscribed to replies of *f*. The server is subscribed to function call requests for its provided function *f*; and publishes the corresponding replies. With this component configuration, the translation from I_M to I_{MZ} is simple, namely all send and receive actions in the client and server models are replaced by write and read actions to the corresponding message types on the publish-subscribe middleware. The time-out is implemented in the connector, which checks whether the read action corresponding to a "wait for reply" action in the model can be executed within the specified amount of time; otherwise it notifies the model that a time-out has occurred. For the integration of a client or server realization, the connector should translate between publish-subscribe messages and real function call requests and replies. For example, when the connector of server realization Z_S receives a request *f_req* via the middleware, it should call the real function *f* of Z_S, after which the result is published on the middleware as *f_rep*. For a realization, the time-out functionality is included in the component realization itself.

Sequential Logic (Electronics)

Many interaction types for electronic components are based on sequential logic, which depends not only on the current state, but also on the previous state. It is typically

Figure 7. I_{MZ} for Function Calls



used to create memory in which values are stored as voltages in the circuits. Latches and flip-flops are well-known sequential circuits that appear in many forms for direct communication between electronic components (e.g., via cables) or for communication between software and electronics (e.g., via memory mapped I/O or distributed I/O). In all these forms of sequential logic, the sending component is able to set a certain value that is stored in the circuit, and the receiving component is able to observe or read this value. Taking the set/reset or SR-latch as a simple example, a sending component can set the SR-latch to active or reset it to inactive (i.e., high or low voltage). In most cases, the state of an SR-latch relates to some internal state of the sending component, e.g., "standby," "ready for next action," or "error." Via the SR-latch, the receiving components can observe this internal state.

Below are some typical sequential logic properties, taking the SR-latch as an example.

- The output value of an SR-latch is continuous (active/inactive) and can only be changed by a set or reset input from the sending component.
- A set or reset input results in an active or inactive latch output, respectively.

Although the SR-latch contains both discrete-event and continuous behavior, which could directly be modeled in hybrid χ (van Beek et al. 2006), we restrict ourselves to the discrete-event version of χ, in which we abstract from the continuous behavior of the SR-latch. A discrete-event model of the SR-latch is shown in Figure 8. The highest level parallel composition (first | | in code) contains the processes of the sending and the receiving component of the *ready_latch*, which indicates whether the sending component is ready for some next action. The sending process first sets the latch output to *false* and later, when it is ready, to *true*. The receiving process waits until the other component is ready, indicated by the latch value *ready*, and continues its behavior (*ready* -> ...). The discrete-event abstraction of latch communication is modeled by adding another process to the model of the receiving component, denoted by the parallel composition on the second level of the model (second | | in code). This additional process is always able to receive new values of the *ready_latch* from the sending component. The variable *ready*, which is used to store the latest latch value, is shared with the other processes of the parallel composition. In this way, only the latest latch value is considered in the behavior of the receiving component.

Figure 8. I_M for SR-latch

```
(   ( ready_latch!false
    ; ...
    ; ready_latch!true
    )
||  ( ready -> ...
    || *(ready_latch?ready)
    )
```

The properties given for the SR-latch are satisfied by the model since the ready variable always has a value (mimicking continuous behavior) and can only be set to true or reset to false by the sending component.

For the SR-latch, the publish-subscribe middleware for the model-based integration infrastructure I_{MZ} is configured with different QoS properties than for the function call interaction type. For function calls, the publish-subscribe middleware acts as a FIFO buffer that does not store its value after delivering it to the receiving component. However, for the SR-latch, it should store the value that is last received from the sending component. This is achieved by configuring the publish-subscribe middleware with the QoS property "keep one message as history."

In the described SR-latch example, only one value is stored (single-address memory). The infrastructure model I_M and its implementation I_{MZ} can easily be extended to represent multi-address memories as used in memory mapped I/O and distributed I/O.

Concluding Remarks

In this paper, we presented an approach on how to deal with infrastructure in the MBI&T method, giving an intuitive indication that the analysis results based on the system model also hold when the models are combined with realizations and the real infrastructure (formal proofs are left as future work). In the presented approach, the behavior of the infrastructure realization I_Z is modeled as infrastructure model I_M. This model I_M is included during system model analysis, and implemented in a model-based integration infrastructure I_{MZ}, using a publish-subscribe middleware, for integration and testing with component realizations. For the examples taken from industrial practice, the transition from synchronous process algebraic models to distributed asynchronous realizations is rather straightforward. The synchronous models provide a good understanding of system behavior and enable verification of properties related to both infrastructural and system behavior. The integration of models and realizations allows fast and cheap system integration and testing several months before real integration and testing, as shown in an industrial case study (Braspenning et al. 2008), in which several system design and integration problems were detected at an early stage and a significant reduction of lead time and costs was achieved. The described approach can be applied to other interaction types in a similar way. The authors would like to thank Albert Hofkamp, Ralph Meijer, Johan Neerhof, and Jan Tretmans for the fruitful discussions and their valuable comments.

References

ASML website. 2008. http://www.asml.com

Baeten, J.C.M. and W.P. Weijland. 1990. *Process algebra*. Cambridge University Press.

van Beek, D.A., K.L. Man, M.A. Reniers, J.E. Rooda, and R.R.H. Schiffelers. 2006. Syntax and consistent equation semantics of hybrid Chi. *Journal of Logic and Algebraic Programming 68,* no. 1-2: 129–210.

Boehm, B.W. and V.R. Basili. 2001. Software defect reduction top 10 list. *IEEE Computer 34*, no. 1: 135–137.

Braspenning, N.C.W.M., E.M. Bortnik, J.M. van de Mortel-Fronczak, and J.E. Rooda. 2008. Model-based system analysis using Chi and Uppaal: An industrial case study. *Computers in Industry 59*, no. 1: 41-54.

Demaine, E.D. 1998. Protocols for non-deterministic communication over synchronous channels. Proceedings of IPPS-SPDP '98: 24–30.

Eugster, P.Th., P.A. Felber, R. Guerraoui, and A. Kermarrec. 2003. The many faces of publish-subscribe. *ACM Computing Surveys 35*, no. 2: 114–131.

Millard, P., P. Saint-Andre, and R. Meijer. 2006. XEP-0060: Publish-Subscribe. Jabber Software Foundation.

Milner, R. 1989. Communication and concurrency. Prentice-Hall.

Mörk, S. 2001. Distributed implementation of a process algebra based programming language for embedded systems. *Nordic Journal of Computing 8,* no. 1: 121–158.

Palamidessi, C. 1997. Comparing the expressive power of the synchronous and the asynchronous π-calculus. In Proceedings of POPL '97: 256–265.

Systems Engineering Group. 2008. Mechanical Engineering Department, Eindhoven University of Technology, http://se.wtb.tue.nl/sewiki/chi

Biographies

N.C.W.M. Braspenning graduated at the Systems Engineering group of the Mechanical Engineering Department of Eindhoven University of Technology, the Netherlands. From 2003 through 2008, he performed his

Ph.D. project titled "Model-based integration and testing of high-tech multi-disciplinary systems."

J.M. van de Mortel-Fronczak graduated in computer science at the AGH University of Science and Technology of Cracow, Poland, in 1982. In 1993, she received the Ph.D. degree in computer science from the Eindhoven University of Technology, the Netherlands. Since 1997 she has worked as an assistant professor at Eindhoven University of Technology, focusing on supervisory machine control.

J.E. Rooda received the M.S. degree from Wageningen University of Agriculture Engineering and the Ph.D. degree from Twente University of Technology, Enschede, the Netherlands. Since 1985 he has been Professor of the Systems Engineering group at the Mechanical Engineering Department of Eindhoven University of Technology, the Netherlands. His research interests are analysis, modeling, and control of manufacturing systems and supervisory machine control.

This work has been carried out as part of the TANGRAM project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under grant TSIT2026.

Boundary Objects as a Framework to Understand the Role of Systems Integrators

Allan Fong, Massachusetts Institute of Technology
Ricardo Valerdi, Massachusetts Institute of Technology
Jayakanth Srinivasan, Massachusetts Institute of Technology

Abstract

The US Department of Defense is facing challenges to develop the capabilities necessary to effectively operate in new operational environments. As a result, these services are seeking to partner with industry members and leverage both government and industry knowledge to develop System of Systems (SoS) that can provide the desired capabilities by integrating legacy systems with new technologies. These large-scale engineering projects require system integrators that can manage not only the technical interfaces but also the organizational ones. This paper proposes a boundary object framework that can assist in understanding the role of these systems integrators by observing changes in organizational interfaces. This framework does so by monitoring the objects and artifacts used at the interfaces.

Introduction

The military is facing new challenges as a result of a tightening spending budget and the need to acquire novel capabilities to operate in new war environments. Meeting these challenges requires integrating legacy systems with developing technologies in a System of Systems (SoS). SoS is defined as having components that are both operationally and managerially independent (Maier 1998). SoS is used to describe both technical and organizational systems. When dealing with the integration of large systems, it is difficult to separate the organizational systems from the technical systems. The interfaces of organizational systems, i.e., the transfer of documentation or requirements from one group to another, are just as important as the interfaces of technical systems, i.e., the exchange of bits, energies, and stresses. The responsibilities of integrating these complex systems now rest on the shoulders of contractors. This leads to the emergence of Lead Systems Integrators (LSIs) as a way to partner with industry members and leverage the technical and managerial knowledge of industry.

Finding a way to understand what systems integrators do is beneficial both operationally as well as for contracting purposes. However, the role and value of the LSI is not well-defined and can be difficult to measure in part because the roles, responsibilities, and boundaries of different stakeholders (customers, integrators, contractors, etc.) involved in a SoS are often blurry. As a result, it is crucial to look at the interfaces within the different constituents of a SoS in order to better define boundaries and assess inter-organizational interactions.

Interfaces amongst organizations occur when there is some kind of formal or informal interaction. These interfaces typically involve the use of some object or artifact that is exchanged between the different stakeholders. These content-carrying objects have been referred to in past literature as boundary objects. This paper applies the boundary object concept to a SoS context and is helpful in understanding inter-organizational interfaces. By understanding the exchange of boundary objects between organizations, one can better appreciate the role and value of a LSI. Although this paper is using the boundary object framework to study SoS inter-organizational interface difficulties, the problems exist in most any complex system development and integration, making this framework widely applicable.

Boundary Objects Literature

Boundary objects were introduced by Star and Griesemer and they defined them as 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 (1989). These objects have different meanings in different communities of practice, but their structures are common enough to more than one community, making them recognizable by a means of translation and interpretation (Star and Griesemer 1989). Objects are generally defined as the artifacts that a person or community works with (Carlile 2002). These objects can be physical objects, such as documents containing diagrams of the system architecture, or electronic objects, such as e-mail. In addition, 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. Table 1 highlights

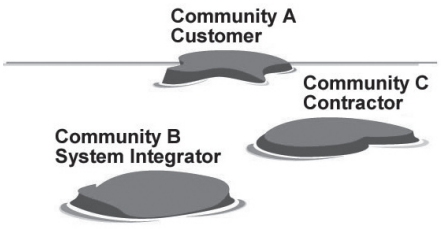
Table 1. Boundary Object Literature

Field	Organization	Boundary object
Social science (Star and Griesemer 1989)	Museum of Zoology	Diagrams California map collecting forms
Design engineering (Henderson 1991)	Engineering firm	Sketches drawings CAD
Product development (Carlile 2002)	Automobile design and manufacturing firm	Drawings automobile parts schedule
Software development (Gunaratne et al. 2004)	R&D facility	Storyboard prototype
Service (Ackerman and Halverson 1999)	Telephone hotline group	Written notes

literature that applied boundary objects to study the interactions between different communities of practices in various fields.

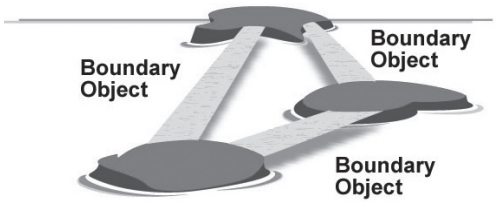
Objects become boundary objects when they are effectively used at the interface of different communities of practice (CoP) to transmit and share information and the context in which the information exists. A CoP is a group across which sense making, understanding and knowledge is shared. More specifically, 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 CoP will develop the same world view or mental model (Brown and Duguid 1998). These CoP have been also referred to as social circles, stakeholders, organizations, etc. Boundary objects essentially exist and are used at the interfaces between these CoP. Figure 1 and Figure 2 represent the purpose of boundary objects. In this example, the separate communities are the customer, system integrator and a contractor. If designed and used properly, boundary objects can connect together what were once separate communities. The boundary object bridges allow the communities avenues to communicate, coordinate and collaborate. This paper considers organizations as CoP and focuses on the use of boundary objects at these community interfaces.

Figure 1. Separate Island Communities



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 distributed. Moreover, these objects can 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 or crossing out certain parts (Swarts 2004). Each style of marking adds an additional layer to the object. The evolutionary characteristic of a boundary object and its ability to carry information and context allow different communities to interface (communicate, coordinate, or collaborate) with each other.

Figure 2. Boundary Objects As Bridges



The following sections further explain the boundary object concepts using three models: a mental, bridge, and characterization model.

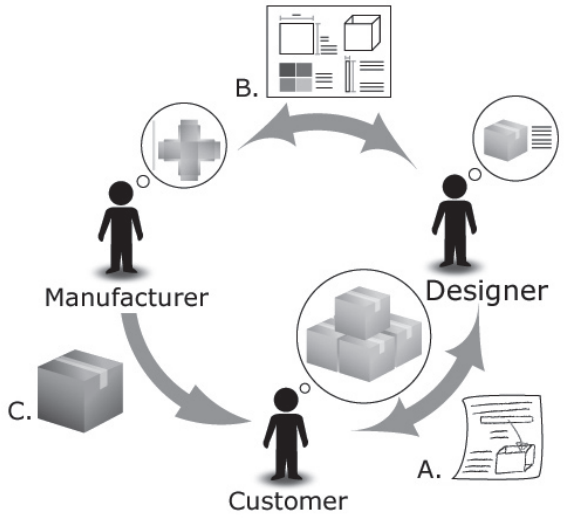
Boundary Object Mental Model

The effectiveness of a boundary object is directly related to how it is translated from tacit knowledge to explicit knowledge (decontextualized) and translated back from explicit knowledge to tacit knowledge (recontextualized) between different communities. For example, a technical drawing can mean different things to a designer and a

manufacturer. The designer might look at the technical drawing and envision how the component fits and functions with other components as an end product. The manufacturer might look at the technical drawing and think about the machining steps necessary to manufacture the component.

The Boundary Object Mental Model helps communities understand how the boundary object is interpreted by other communities. It increases understanding of the context in which these objects will be used and is very important for the system integrator. Figure 3 is a depiction of different mental models during a simplified design/manufacturing process.

Figure 3. Boundary Object Mental Model



The process starts in figure (A) between the customer and the designer. The customer desires a specific component and has a mental model of what that component is going to be used for. The customer needs to translate his mental model to the designer. He needs to decontextualize his idea into a transferable form for the designer. To accomplish this, a boundary object, in this case a sketch and description of the component, is created by one party and interpreted by the other. When the designer looks at the drawing, he will translate it to a specific mental model focusing more on the technical properties of the component rather than its eventual use by the customer.

The designer now needs to translate his model to the manufacturer, as shown in (B). To do this, the manufacturer and designer have to work together to create a boundary object, a technical drawing, that both parties can understand. The object contains the decontextualized knowledge from the designer that can be recontextualized by the manufacturer. Nevertheless, when a manufacturer looks at the drawing, he will focus on the assembly aspect of the component.

Once the component is manufactured, it becomes a boundary object, as shown in Figure 3(C). The arrow could potentially be unidirectional, in which case, the customer does not provide feedback to the manufacturer if changes are needed. If the customer is not satisfied, he will need to talk to the designer again. Although this is a simple model, it highlights a problem area that exists between the manufacturer and customer. The information decontextualized into the final product will not be successfully recontextualized by the customer if the part is not exactly what the customer desires.

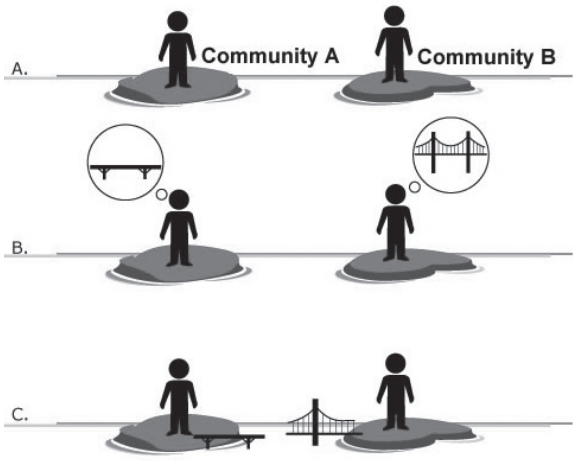
Boundary Object Bridge Model

An additional role of a boundary object is to bridge the understanding and communication gaps between different communities. A boundary object, a bridge, must be developed with input from all of the sides. Logically, the more communities that the boundary object needs to connect, the more complicated the boundary object becomes. It is important for there to be effective communication between all of the parties involved with the development of a boundary object.

The types of bridges used will be specific to the gaps they need to connect. The solution must match the need. Sometimes the best solutions are the simple and cheapest ones. Other times, expensive bridges must be built.

The bridge model concept is illustrated in Figure 4.

Figure 4. Boundary Object Bridge Model



In Figure 4(A), community A and community B are on two different islands. In Figure 4(B), they both want to construct a bridge to close the gap between them but without communication they develop different solutions to the same problem. This lack of communication leads to both sides constructing different bridges as shown in

Figure 4(C). Problems will occur when they try to connect the two bridges. This will lead to rework and wasted resources. However, if both groups start with a common vision they will be able to construct a successful bridge between them.

Figure 5. Sharing Boundary Objects

In Figure 5(D), the bridge drawings sent back and forth between the stakeholders are the boundary objects that connect both parties.

Figure 6. Connected Islands

A successful bridge must include the input of and be developed by both stakeholders as shown in Figure 6.

Boundary Object Characterizations

This section will discuss six different attributes for boundary objects: type, functionality, utility, information granularity, context and familiarity. This boundary object model is being developed and validated through case studies. These axes may evolve as this research and similar efforts continue.

Type

Boundary objects can be distinguished into two types of objects: virtual and physical. Virtual boundary objects are those that exist in bytes and bits. They are stored in computers, databases, etc and are transferred electronically. Examples of virtual boundary objects are e-mails, websites, and electronic databases. Physical boundary objects are objects that are tangible and can be physically manipulated.

Functionality

Star and Griesemer categorized boundary objects into four functional categories: repositories, ideal type, coincident boundaries, and standardized forms. Repositories are ordered collections of objects such as a library or database. Ideal types are abstractions from different domains and may be open to a fairly broad spectrum of interpretation. Ideal types include diagrams, drawings, and clay models. Coincident boundaries are common objects which have the same boundaries but different internal contents (Star and Griesemer 1989). An office building is an example of a coincident boundary because representatives from different organizations can all work within the same physical boundary. Lastly, standardized forms are objects that provide different communities with a common way to communicate. Standardized forms include forms for clearance procedures and proposal submissions.

Utility

The utility of the boundary object is the degree of cognitive usefulness the user finds in the object. This attribute measures the degree in which the object will influence the user's task.

Granularity

Granularity describes the level of detail of the information in the boundary object. In many cases, objects carry vague or misleading information. Objects that use ambiguous terms, such as "very much" or "too little," can lead to confusion between communities of practices. Furthermore, an object can include different amounts and types of information. For example, the financial record of a company can be presented in a large excel chart with all of the spending and earning numbers or it can be presented in a word document that summarizes all the numbers.

Context

The context of the boundary object describes how well it addresses the different social contexts and mental models of the user groups. These differences can lead to understanding gaps, which were addressed as attributes of the coordination and collaboration interface. Some communities may be able to understand each other better than others. Their mental models are more aligned and, in these cases, it may be easier to bridge the understanding gaps.

Familiarity

The manner in which boundary objects are used also depends on the familiarity of the stakeholders involved in the interaction. Previous partnerships and contractual agreements are examples of how stakeholders can increase their familiarity with each other. These relationships can affect the trust between the stakeholders. Using the boundary object implies a level of trust between the parties involved. Trusting what is represented in the object and trusting the organization that sent it is essential for collaborative interfaces. If the object clearly represents

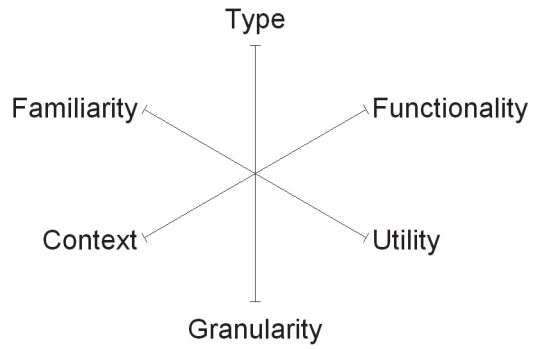
all the information needed between two stakeholders, but one stakeholder doesn't trust the other stakeholder, then the former user will probably be hesitant to use the information.

An additional characteristic of boundary objects is their need for synrochnization (an attribute not mentioned in this paper). A change in the information in one object must propagate to other tightly coupled objects. The appropriate configuration management processes must be in place in order for this to be effective.

Boundary Object Characterization Model

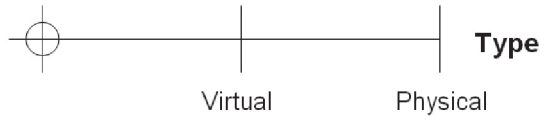
The Boundary Object Characterization Model (Figure 7) applies previous boundary object literature to characterize boundary objects based on their type, functionality, utility, granularity, context, and familiarity between the user groups, as shown in the following figure. This paper proposes the model as a novel method to characterize the boundary objects used at an organizational interface. By considering the objects used at current interfaces, one can create new interfaces or modify existing ones to create more capabilities in the system.

Figure 7. Boundary Object Attributes

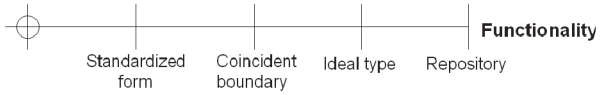


The axes for the boundary object attributes are described next.

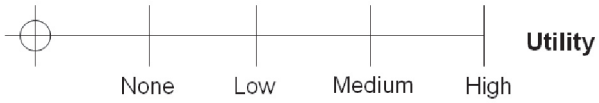
Type



Functionality

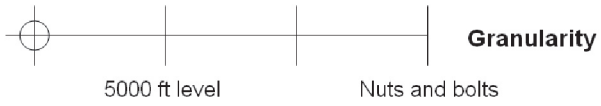


Utility



Granularity

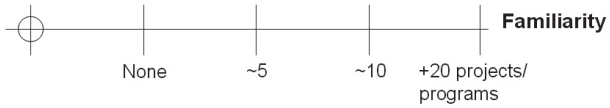
This is the level at which information is represented from very high conceptual level (the 5000ft level) to the nuts and bolts specifics.



Context

Familiarity

This can range anywhere from no previous relationship to past partnerships on more then 20 different programs or projects.



The Boundary Object Characterization Model gives a numerical representation 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. Although the Boundary Object Characterization Model is based largely from past literature involving interactions within an organization, this research aims to take the understanding of boundary objects within an organization and apply it to inter-organizational interfaces through case studies. Furthermore, this framework will be a useful tool for systems integrators in understanding and diagnosing organizational interfaces failures.

Model Application

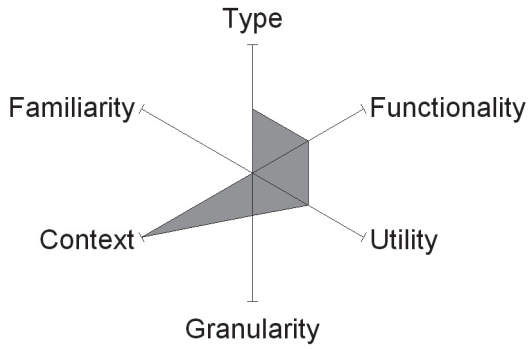
In a project as large and complex as the Army's Future Combat System, there are many interfaces between different communities of practices. At each of these interfaces exists some kind of interaction; usually involving a significant amount of information and knowledge exchanges. The role of boundary objects in these interactions is to

provide a characterization of the artifacts that exist in the collaborative environment.

The Boundary Object Characterization Model enables the analysis of organizational interfaces by characterizing the objects used at these interfaces. The following example is of how this model can be applied to organizational interfaces between an LSI and a contractor that have never worked together before.

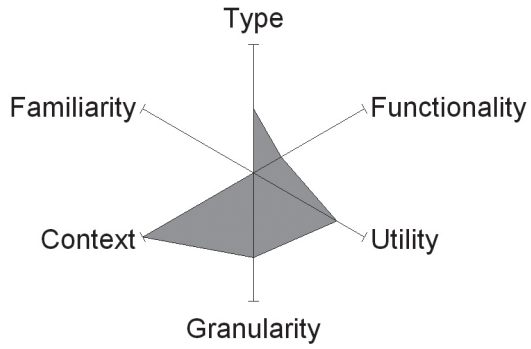
First, the LSI posts general information about an upcoming program on their website. This is a virtual boundary object and is used a few times because the information on the website is still general and high level. However, the website provides a lot of context and program background, as shown in Figure 8.

Figure 8. Website Boundary Object



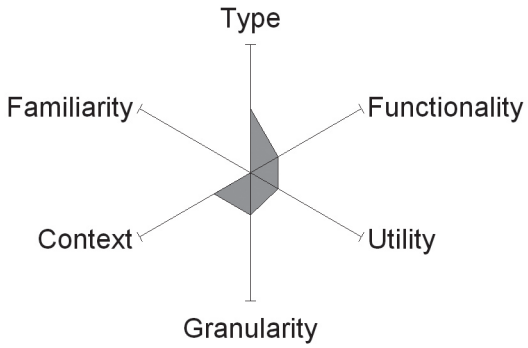
As the LSI receives more requirements and directions from the customer, they will solicit proposals for companies who are interested. This request for proposals is also done electronically in a standard format. The request will carry a lot of context and more information than just the website, as shown in Figure 9.

Figure 9. Description of Proposal Object



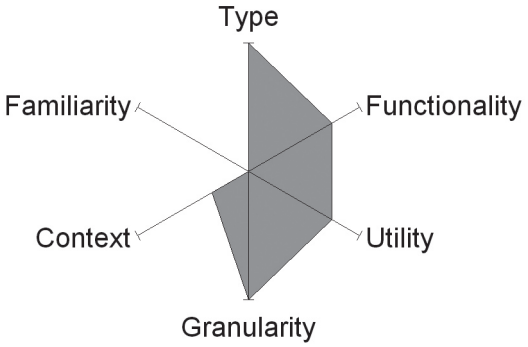
After the contractor is selected, they will have to provide bi-weekly presentation updates. These presentations are high level and use PowerPoint. The PowerPoint slides are used only once and do not carry a lot of context because the context is communicated verbally, as shown by Figure 10.

Figure 10. Presentation Object



Lastly, a physical prototype model is used between the contractor and LSI. There is a lot of information imbedded in the model but does not carry much context, as shown in Figure 11.

Figure 11. Description of Prototype Object



Although there are a lot of interfaces besides the ones previously mentioned, this example shows that this model can be used to capture the type of interface between organizations. This example also shows that boundary objects change as relationships and interfaces between organizations evolve. Different types of boundary objects are represented by the different shaded shapes. Additional research will be done to see what the correlations are between the shape of the graphs to the type of interfaces and cost of the object. The evolution of boundary objects can assist in understanding organizational system dynamics. A further expansion of this concept will be included in following papers.

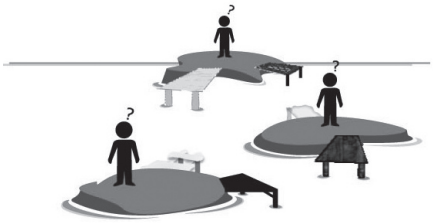
Implications for System Integrators

The most value or leverage in constructing a SoS is at the interfaces (Maier 1998) and it is at these interfaces that the significance of boundary objects is realized. 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. As a result, the role of a systems integrator is, as the name implies, to integrate various systems together by managing the interfaces. 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.

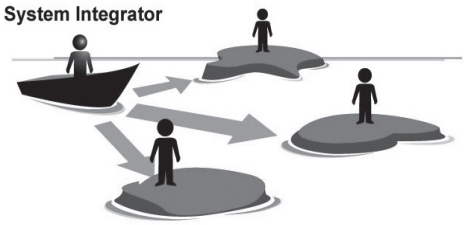
In a SoS with no integrator, the different organizations can be thought of as disconnected islands. Figure 12 is similar to the bridge model previously described. Before the bridge boundary objects are constructed, the different communities will have to work together or else they might end up with different bridge designs incapable of interfacing.

Figure 12. Communities with Incompatible Interfaces



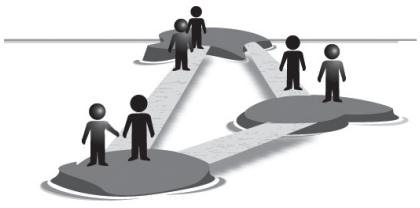
The systems integrator needs to work within all the communities and become the vital link that connects them, as shown in Figure 13. For example, the LSI for the Future Combat System uses a software collaborative environment to organize all of their project related files.

Figure 13. System Integrator Involvement



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.

Figure 14. Increase Collaboration and Value



A 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 systems 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 begins, the system integrator must make sure all the communities can understand the information they receive from each other. If each community spoke a different language, the system integrator must provide some method for translating the languages. The systems integrator must be able to address failures in communication, coordination and collaboration between different organizations.

Certain types of boundary objects will be more effective in some environments as compared to others. Boundary objects can be used to measure the fluidity and flexibility of different constituent systems. This paper provides a quantitative model for understanding correlations between inter-organizational interfaces and the boundary objects used at these interfaces.

Social integration is as important as technical integration and this boundary object research highlights the human aspect of interfacing within System of Systems.

Furthermore, this framework is not only limited to SoS. Most large complex systems face the same inter-organizational interface problems described in this paper and can be studied to assess the validity of the framework. Additional developments of the boundary object framework can also provide a tool to monitor and measure the integration of different complex systems.

Conclusion

The interfaces within a System of Systems (SoS) are where the benefits of a SoS come from, making the role of system integrators extremely important. Although this role is essential to the SoS, it is not well defined. This paper proposes boundary object models to analyze the role of the system integrator by focusing on how stakeholders in a SoS interact. A second-order benefit of this approach is the ability to predict possible failures in a

program as indicated by the poor use of boundary objects between stakeholders.

There is much work that can be done in this area of research. For instance, the impact of open standards can be evaluated in the context of boundary object attributes.

In parallel to the improvements of this framework, we hope that this paper will open the door to a new way of thinking when valuing the role of system integrators.

Acknowledgments

This research would not have been possible without the support of the Lean Advancement Initiative consortium members and financial support from the Aerospace Corporation. The authors of this paper would also like acknowledge and thank Diana Fong, for designing the graphics used in this paper. Furthermore, acknowledgements and thanks are extended to the individuals who participated in this study.

References

Ackerman, Mark S., and Christine Halverson. 1999. Organizational memory: Processes, boundary objects, and trajectories." Proceedings of the 32nd Hawaii International Conference on System Sciences. Hawaii.

Brown, John S. and Paul Duguid. 1998. Organizing knowledge. *California Management Review* 40: 28-44.

Carlile, Paul R. 2002. A pragmatic view of knowledge and boundaries: Boundary objects in new product development." *Organization Science* 13: 442-455.

Carlile, Paul R. 2004. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science 15*: 555-568.

Gunaratne, Junius, Beatrice Hwong, Christopher Nelson, and Arnold Rudorfer. 2004. Using evolutionary prototypes to formalize product requirements. Siemens Corporate Research, NJ.

Gupta, Amar. 2003. Role and importance of lead system integrator in context of new air operations centers. White paper MIT Sloan School of Management.

Henderson, Kathryn. 1991. Flexible sketches and inflexible data bases: Visual communication, conscription devices, and boundary objects in design engineering. *Science, Technology, and Human Value 16*: 448-473.

Maier, Mark W. 1998. Architecting principles for systems-of-systems. The Aerospace Corporation CH 1-460.

Sapsed, Jonathan, and Ammon Salter. 2004. Postcards from the edge: Local communities, global programs and boundary objects. *Organization Studies* 25: 1515-1534.

Spurlock, Darren M. 2005. Space exploration systems integration. 1st Space Exploration Conference: Continuing the Voyage of Discovery Proceedings.

Star, Susan L., and James R. Griesemer. 1989. Institutional ecology, "translations" and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science 19*: 387-420.

Swarts, Jason. 2004. Textual grounding: How people turn texts into tools. *Journal Technical Writing and Communication 34*: 67-89.

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 Advancement Initiative. Allan graduated from Columbia University in 2005 with a B.S. in mechanical engineering.

Ricardo Valerdi is a Research Associate with the Lean Advancement Initiative at MIT. He is currently the research lead for the Enterprise Metrics cluster. Ricardo received his Ph.D. 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 Engineer with the Lean Advancement 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.

Addressing System Boundary Issues in Complex Socio-Technical Systems

Joseph R. Laracy, Massachusetts Institute of Technology

Abstract

Systems engineering researchers are familiar with a variety of challenges associated with doing foundational research in complex socio-technical systems. Some foundational issues have been avoided by focusing on applied research questions and ignoring the "socio" of the engineering system under development. Considerations of large-scale engineering systems often present a dilemma of where to draw the line between a system and its environment. How are social, political, economic, and institutional issues addressed? The lack of suitable methodologies for understanding the interface between a technical system and the human and organizational it exists within is a stumbling block. The author suggests a way ahead drawing on the ancestral disciplines of systems science. This approach led to the development of a system safety engineering methodology, System-theoretic Accident Models and Processes (STAMP), which has had significant impact on industry and the practice of safety engineering.

Introduction

Researchers and practitioners in the field of systems engineering occasionally refer to the systems they develop as *socio-technical*.

> The socio-technical concept arose in conjunction with…several projects undertaken by the Tavistock Institute in the British Coal Mining Industry. The time [1949] was that of the postwar reconstruction of industry… The second project was led, through the circumstances described below, to include the technical as well as the social system in the factors to be considered and to postulate that the relations between them should constitute a new field of inquiry. (Trist 1981)

The inclusion of human factors in the design of engineering systems was revolutionary at that time and still is today in some academic and industry circles. In 1930, MIT President Karl Compton initiated a movement to make the practice of engineering more scientific, thereby initiating the approach of *engineering science*. Engineering science—applied physics, chemistry, and mathematics—proved to be very successful in the Second World War. The development of Radar is often cited as a product of the engineering science approach (Mindell 2004). Immediately following the war, the creation of the National Science Foundation revived the question of what it meant to do basic research in an applied field such as engineering (Kline 2000).

As systems continued to grow in size and complexity, the aerospace industry responded with what is now called systems engineering. The program for America's first ICBM, the Atlas missile, served as a test-bed for this new approach to interdisciplinary engineering system design. The Semi-Automatic Ground Environment (SAGE) air defense system, which enabled the North American Aerospace Defense Command (NORAD) to track, and if necessary coordinate a military response to Soviet strategic bombers, also made use of early systems engineering practices (Hughes 2000; Hughes 1998).

In parallel, system theorists in academia were considering many of the same concepts as industry engineers such as feedback, dynamics, flows, block diagrams, human-machine interaction, signals, simulation, and computers (Mindell 2004). However, as Kroes et al. point out, both groups encountered a serious problem:

> The field of systems engineering has inherited a conceptual problem from systems theory. Just as systems theory since its beginnings has been plagued by the question how to separate a system from its environment or context, the field of systems engineering has been confronted with a similar question about engineering systems. How are the boundaries of [engineering] systems to be drawn? What belongs to the [engineering] system under consideration and what to its environment? *For engineering systems this problem manifests itself conspicuously with regard to the status of non-technical elements, such as social, political, economic and institutional ones.* [emphasis added] To what extent are these, or ought these elements to be considered to belong to engineering systems or to the environment or context? (Kroes 2004)

Unfortunately, engineering science lacked the tools to address these fundamental questions in the new field of systems engineering.

The Boundary Problem

Catastrophic failures are associated with ignoring social, political, economic, and institutional elements. Mindell writes:

> It is highly significant that the Columbia Accident Investigation Board identified 'history and culture' as a major contributing cause of the accident. History and culture are not mysterious, inhibiting forces that act on the technological development; they are just as integral to technology as are Newton's laws and Fourier transforms. (Mindell 2004)

Another great defeat of the systems approach is associated with Robert McNamara's "Whiz Kids." "Through systems analysis, McNamara and his staff felt empowered to replace the complexity of real life with simplified models that lent illusory precision by their quantitative bases." (Jardini 1998) By dismissing many human variables and approaching the Vietnam War only as a national defense production problem, decisive factors in the outcome of the conflict, such as the fighting will of the North Vietnamese, were ignored.

Civilian problems such as housing, health care, education, poverty, and transportation were also studied with the systems analysis approach. Programs that modeled human factors and left room for compromise and negotiation were much more successful than those that left them out. For the unsuccessful programs, Mindell points out that "in retrospect, the engineers would often point to the detrimental effects of politics, which stifled or derailed their projects. But in doing so, they pointed to the limitations of their models, which excluded politics and the social world as *external* variables." (Mindell 2004)

Clearly, the "socio" of socio-technical systems cannot be ignored. The work of Thomas Hughes is useful in considering large technical systems as a seamless web of social and technical elements where one distinguishes between physical artifacts, organizations, scientific components, legislative artifacts, and natural resources (Bijker ed. 1987). This view leads systems engineering researchers to ask the question of where to draw the boundary of the system and its environment. Furthermore, if social elements are considered, how are they to be analyzed?

One of the most conspicuous problems facing the systems engineer is the lack of formal education or on-the-job training to rigorously analyze the social forces that influence a system. An ABET accredited program does not require coursework in designing stakeholder surveys, conducting human experiments (human factors engineering), designing meaningful interviews, and other useful skills for engineering large scale, complex systems. Systems engineering researchers working on safety problems at MIT are assisted in this regard by the System Safety Working Group. The group has scholars in fields such as aerospace engineering, social psychology, computer engineering, organizational behavior, civil engineering, industrial relations, physics, and of course systems engineering. In this endeavor, a careful balancing act must be carried out. The systems engineer should acknowledge that "a systems approach is centered around the human being" and "the efficient design of systems is influenced decisively by the people who have to operate them" (Jenkins 1971). Nevertheless, he must also appreciate the field's scientific roots in dynamical systems theory, control theory, and biology (Emes 2006).

In essence, the problem comes down to methodology. How can the techniques of engineering science be connected with a modern understanding of human decision making, organizational behavior, and institutional inertia?

A Way Ahead—The Ancestral Disciplines

The good news is that many people have made significant progress at answering this question. The ancestral disciplines of systems science have much to offer 21st century systems engineers. Unfortunately, the term "systems thinking" has been so abused and misused that it has been reduced in many circles to a consulting buzzword. However, true systems thinkers—or those that take a systems approach—should expose themselves to the richness of:

1. General System Theory
2. Cybernetics
3. System Dynamics
4. Complex Adaptive Systems
5. Control Theory

The ancestral disciplines are useful in two ways:

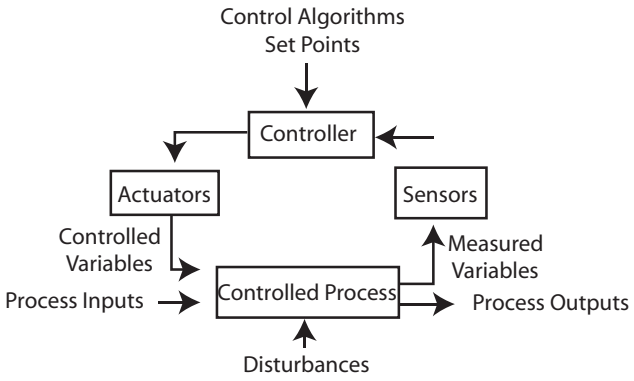
1. Scholars in the respective fields have confronted the human-machine problem directly and quite successfully.
2. New theories of socio-technical systems can be developed by creatively integrating the techniques of the ancestral fields.

In his *General System Theory*, Von Bertalanffy presents the concept of an open system: "An open system is defined as a system in exchange of matter with its environment…" (von Bertalanffy 1969). The concept of an open system is an important one in applied science, because often the pure sciences (i.e., chemistry) make an assumption of a closed system—one isolated from its environment. This assumption has been implicitly imported into engineering design mental models. However, the large scale, complex systems that are the concern of the systems engineer are inherently open.

Often, engineers draw the boundary between system and environment in their models when they encounter variables that they cannot control. However, abstracting away variables that are beyond one's control does not mean they are being handled correctly. Cybernetics offers many insights into modeling human-machine systems (Ashby 1956; Wiener 1965). Cybernetic systems are inherently purposeful, goal-directed systems. The most fundamental model of control is shown below in Figure 1.

Perturbations to the controlled process change the process in such a way that the sensors report the change to the controller which issues orders to the actuator to move the system toward the goal condition. While this model may seem trivial, it is useful to look deeper and realize that the entire model can be inverted. The environment has its own goals. The "external" disturbances of the environment attempt to impose its own set points for the process. In a symmetric scenario, such a process will never reach a stable equilibrium (Heylighen 2001). Through the IEEE Systems, Man, and Cybernetics Society as well as through some European faculties, cybernetics research continues to this day, albeit not nearly as pervasively as its founders would have hoped. With the closing of Heinz von Foerster's Biological Computer Laboratory at the University of Illinois, and other similar cybernetic research communities, the field deliquesced into computer science, decision and control engineering, artificial intelligence, robotics, and bioengineering (Hutchinson 2006).

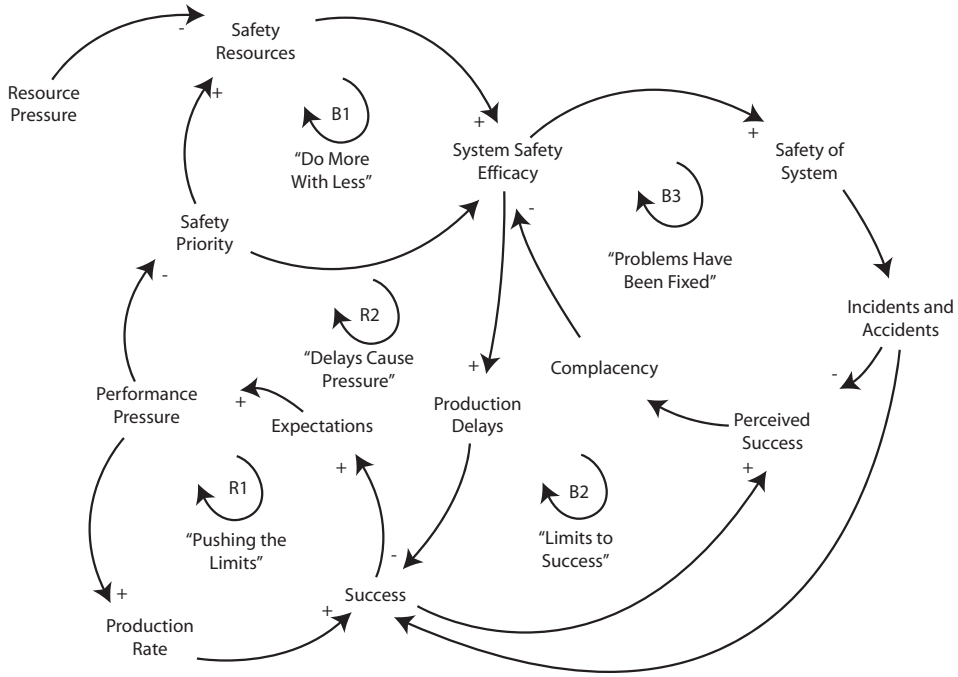
Figure 1. A Feedback Control System
Image Source: Leveson 2002



Jay Forrester's System Dynamics (Forrester 1961) builds on the ideas of General Systems Theory and Cybernetics. von Bertalanffy's notion that complex systems can be modeled by systems of nonlinear differential equations and Wiener's notions of feedback and control are central themes of System Dynamics modeling. System Dynamics addresses concepts such as dynamic complexity, bounded rationality, flawed mental models, policy analysis, nonlinear (unintuitive) behavior, causal loops, delays, stocks and flows, and many concepts relevant to socio-technical system modeling (Sterman 2000).

System Dynamics does not distinguish between "hard" and "soft" variables as is the case with traditional engineering models. For example, a system safety engineer can develop a technical model of the physical system (i.e. a nuclear power plant) as well as the supporting human and organizational factors. The model shown in Figure 2, developed by Dulac and Leveson, captures important dynamic phenomenon such as "pushing the limits," "doing

Figure 2. High Level Abstraction of a System Dynamics Model for Safety in Operations
Image Source: Dulac 2005



more with less," "delays cause pressure," and other feedback loops encountered in real world complex systems.

Another ancestral discipline relevant to this discussion is the area of Complex Adaptive Systems (CAS). CAS such as the human brain, ecological systems, artificial neural networks, and some parallel distributed computing systems are characterized by the emergence of complex behaviors "as a result of often nonlinear spatio-temporal interactions among a large number of component systems at different levels of organization" (Chan 2001). Attributes of CAS include a reliance on distributed control, sensitivity to interconnectivity of components, co-evolution of the system with its environment, sensitivity to initial conditions in the case of mathematical chaos, and avoidance of equilibrium conditions. Engineering systems that exhibit properties of CAS cannot be separated from their environment. Chan states:

> CAS are dynamic systems able to adapt *in* and evolve *with* a changing environment. It is important to realize that there is no separation between a system and its environment in the idea that a system always *adapts to* a changing environment. Rather, the concept to be examined is that of a system *closely linked with* all other related systems making up an ecosystem. Within such a context, change needs to be seen in terms of *co-evolution with* all other related systems, rather than an *adaptation to* a separate and distinct environment. (Chan 2001)

Therefore, it is important for systems engineers to identify whether their system may exhibit CAS properties, and if so, ensure that their models acknowledge the intimate connection between the engineered system and environment. Agent-based modeling has been shown to be a valuable technique for understanding complex adaptive systems (Krenzke 2006).

Finally, control theory must be re-examined for its applicability to socio-technical systems. While many engineers have taken courses in this area and some have developed specialization in it, engineers tend to assume that the central ideas are limited to purely electrical and mechanical systems. Notions of feedback, stability, controllability, observability, and robustness can be applied creatively to improve the design and analysis of socio-technical systems.

System theorists generally acknowledge three types of structural organization. *Organized simplicity* is exhibited in traditional deterministic systems that can easily be decomposed into subsystems and components such as in structural mechanics. Systems that exhibit *unorganized complexity* on the other hand cannot be decomposed into parts. However, statistical techniques are applicable because of the regularity and randomness that characterize the system. The Law of Large Numbers becomes applicable and average values can be computed such as in statistical mechanics. The "new" complexity, *organized complexity*, describes systems that are too complex to be modeled with analytic reduction but not random enough to be modeled using statistics (Owens 2006). Figure 3 shows the relationship between the three types of organization.

Figure 3. System Organization and Complexity.
Image Source: Weinberg 1975

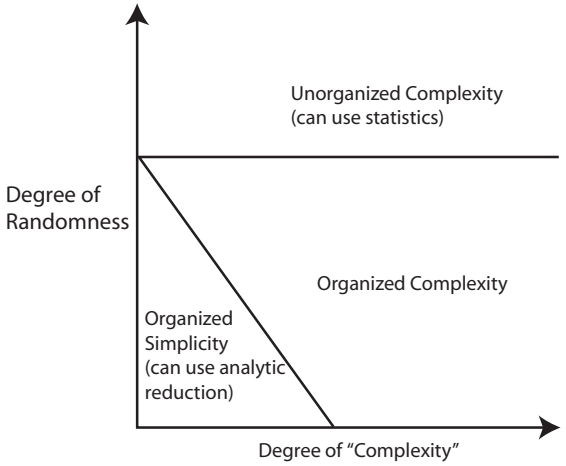


exhibit strong, non-linear interactions and coupling between subsystems and components. Therefore, these systems must be studied holistically. Two underlying concepts provide insight into these complex systems: emergence & hierarchy and communication & control.

Abstractions for complex systems often involve layers. In the case where hierarchy exists, the level of organization increases as one moves toward higher layers. Additionally, the step from level n to $n + 1$ yields new properties that are not discernable at level n. This phenomenon is referred to as emergence, or emergent properties (Leveson 2002). As the next section will illustrate, reliability techniques that are effective for systems exhibiting organized simplicity are not necessarily applicable to systems exhibiting organized complexity.

System-Theoretic Accident Models and Processes (STAMP)

Traditional models of accident causation are rooted in a chain-of-events perspective. Whether part of a preliminary hazard analysis or an accident reconstruction activity, the engineer attempts to understand the potential or actual accident by identifying the events or faults that could initiate the accident. Such fault and event trees are usually part of a method called probabilistic risk assessment (PRA). The goals of PRA are to estimate both the likelihood

and severity of a risk. PRA was developed in the mid 1970s to improve nuclear power plant safety. Professor Norm Rasmussen of MIT chaired the Reactor Safety Study that was the first real probabilistic risk assessment (Apostolakis 2000).

A probabilistic risk assessment is a four step process:

1. Identify undesirable events.
2. Identify accident scenarios (sequences of events).
3. Estimate the probability of each scenario either based on statistical testing data, or expert judgment if scenarios are rare.
4. Rank the accident scenarios according to likelihood.

The framework yields a probability for each undesirable event identified in stage 1.

PRA turned out to be very successful for assessing risks in nuclear power shut-down systems. Such systems were historically very simple, electro-mechanical systems designed to minimize unnecessary complexity and used proven analog electrical technologies. PRA carries with it a number of important assumptions:

1. The events or faults in the trees are collectively exhaustive—all possible events are identified.
2. The events or faults in the trees are mutually exclusive —they cannot occur simultaneously.
3. The probability of each scenario is accurate enough to be useful to decision makers.

In the reactor shut-down system, nuclear engineers with decades of experience can probably develop trees that satisfy the first two assumptions due to their intimate knowledge of reactor design and operation. Furthermore, component technologies such as electrical relays could be extensively tested in the laboratory to compute reliability metrics such as mean time between failures (MTBF).

However, when complex systems like the Space Shuttle are considered, serious questions arise regarding the appropriateness of PRA. For instance, how does software change the picture? How can the MTBF of unique digital electronics be estimated? How many events or faults must be accounted for? Herein lies the problem of applying PRA to software-intensive systems. Software does not wear out and fail; it only implements a set of requirements that may or may not be correct. Subjective probability (expert judgment) must be used when thousands of laboratory MTBF tests cannot be carried out. If a spacecraft computer has 128 MB of memory, or 2^{30} bits, then it has $2^{\text{number of bits}}$ or $2^{2^{\wedge}30}$ states. Clearly, each state cannot be analyzed.

Before the Space Shuttle Challenger disaster, NASA headquarters reported the probability of a failure with loss of vehicle and human life as 10^{-5} (Feynman 1986). Before the Space Shuttle Columbia disaster, the reported probability was 1/250 (Stamatelatos 2002). According to NASA space operations spokesman, Allard Beutel, the post-Columbia figure is now 1/100 (Scottberg 2006).

Formal methods have also been proposed as a solution to the software safety problem. However, the complexity of formal specifications can quickly become unmanageable in large systems. In fact, it is possible for a formal specification to be longer and more error prone than the source code it specifies (Leveson 2002). Additionally, a graduate degree in applied mathematics (formal logic) is required to rigorously apply formal methods.

A new model of accident causation is needed that recognizes the influence of software in the dynamic nature of accidents as well as the human and organizational factors. According to Leveson, "The hypothesis underlying the new model, called STAMP, is that *systems theory is a useful way to analyze accidents*, particularly system accidents" [emphasis added] (Leveson 2004). Component failures associated with hardware reliability engineering are not the only causes of accidents. Accidents often occur in complex systems when external disturbances or dysfunctional interactions among system components are not adequately handled by the *control system.* Inadequate control of safety constraints on system development and operation is the fundamental problem. "Safety then can be viewed as a *control problem*, and safety is managed by a *control structure* embedded in an *adaptive socio-technical system*" [emphasis added] (Leveson 2004). As shown in Figure 4, STAMP utilizes ideas from the ancestral systems science disciplines as well as traditional systems engineering.

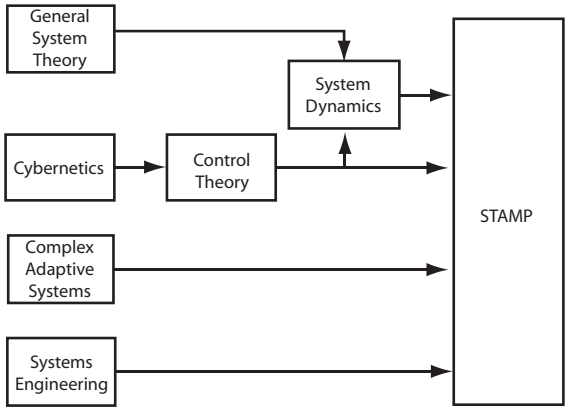
A STamP-based Analysis, or STPA, has five steps.

1. Identify the system hazards.
2. Identify system-level safety constraints.
3. Define the control structure.
4. Identify instances of inadequate control that could lead to a hazard.
5. Model the behavioral dynamics of the system with System Dynamics.

An example of system-level hazards for an air traffic control system is given in (Leveson 2002):

1. Controlled aircraft violate minimum separation standards (NMAC).

Figure 4. Ancestral Roots of STAMP



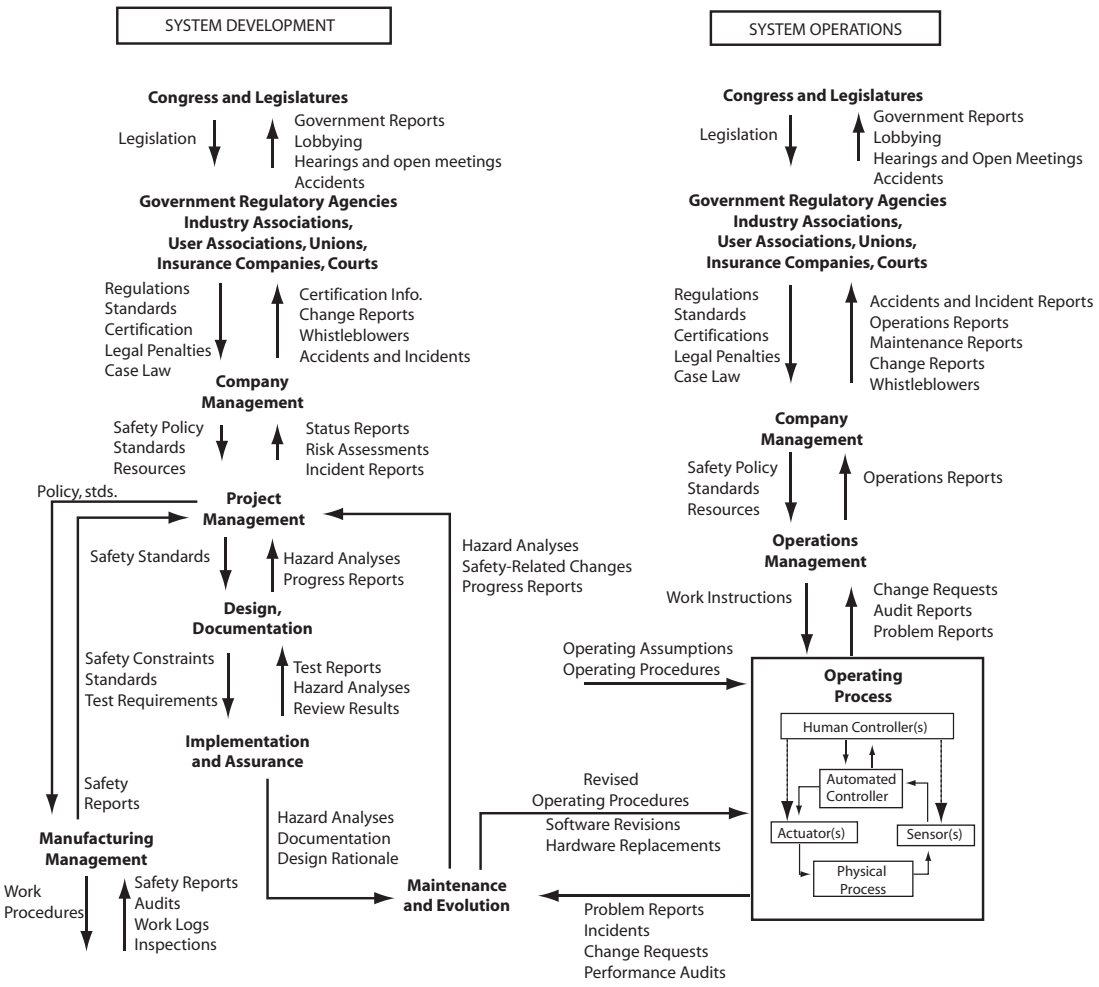
2. An airborne controlled aircraft enters an unsafe atmospheric region.
3. A controlled airborne aircraft enters restricted airspace without authorization.
4. A controlled airborne aircraft gets too close to a fixed obstacle other than a safe point of touchdown on an assigned runway (CFIT).
5. A controlled airborne aircraft and an intruder in controlled airspace violate minimum separation.
6. A controlled aircraft operates outside its performance envelope.
7. An aircraft on the ground comes too close to moving objects or collides with stationary objects or leaves the paved area.
8. An aircraft enters a runway for which it does not have a clearance.
9. A controlled aircraft executes an extreme maneuver beyond its performance envelope.
10. Loss of aircraft control.

It is important to note that this approach is "top-down" as opposed to the "bottom-up" approaches like event trees that must identify every undesired event and trace it up to the unsafe state, or hazard. Consistent with other systems engineering activities, hazards are decomposed to the point where they can be managed. This top-down approach produces a manageable number of hazards, rather than an unmanageable number of undesirable events.

Safety constraints are simply negative requirements. For example, the constraints for hazard 3 are "(a) ATC must not issue advisories that direct an aircraft into restricted airspace unless avoiding a greater hazard. (b) ATC shall provide timely warnings to aircraft to prevent their incursion into restricted airspace." (Leveson 2002) System safety engineers are very familiar with writing requirements so safety constraints are a natural extension.

Utilizing the principles of control theory, a control structure is developed for the socio-technical system. Constraints are assigned to individual components in the structure, and control actions are defined to implement the constraints. The generic model of control is provided in Figure 5.

Figure 5. Generic Model of Socio-technical Control
Image Source: Leveson 2002



Many accidents are not associated with component failure. Instead, they are the result of a slow degradation of the safety culture supporting the system and the development or operations enterprise. Systems migrate toward a state of greater risk in such a way that the evolution is not appreciated until an accident occurs. This notion of evolution is well understood with the techniques of Complex Adaptive Systems.

Identifying instances of inadequate control is a process of studying the control structure for ways that feedback, or more generally control, could be disrupted. A hierarchical taxonomy of such risks has been identified with the following three types at the highest level:

1. Inadequate Enforcement of Constraints (Control Actions)
2. Inadequate Execution of Control Action
3. Inadequate or Missing Feedback

This idea of studying feedback in socio-technical systems originates in the Cybernetics movement.

Finally, System Dynamics modeling is used to understand the behavioral dynamics of the system (Dulac 2005). Inadequate controls previously identified can be prioritized by quantitatively assessing their impact on key system safety variables. Additionally, response mechanisms can be tested, and their effectiveness judged (Laracy 2006).

Conclusion

Modeling large scale, complex systems is not an easy task. Addressing boundary issues between the technical system and the environment are particularly difficult. Often interdisciplinary expertise is needed to address the spectrum of challenges present in socio-technical systems. At MIT, the System Safety Working Group's unifying methodology, STAMP, draws from the ancestral systems sciences. By studying the ideas of the earlier systems scientists and developing new theories of socio-technical systems from them, systems engineers can hope to live up to the standards of General Bernard Schriever of the Air Force Research and Development Command. General Schriever once remarked that a systems engineering contractor should be staffed by "'unusually competent' scientists and engineers to direct the technical and management control over all elements of the program" (Hallam 2001).

Acknowledgements

I would like to thank my advisor, Professor Nancy Leveson, and the Columbia System Safety Working Group for sharing their ideas with me. I also appreciate the review and feedback received from my colleagues, Brandon Owens and Justin Colson, on this paper.

References

Apostolakis, G. 2000. The nuclear news interview—Apostolakis: On PRA. *Nuclear News*: 27-31.

Ashby, W. R. 1956. *An introduction to cybernetics.* London: Chapman and Hall.

Bijker, W., Thomas P. Hughes, and Trevor Pinch, eds. 1987. *The social construction of technological systems: New directions in the sociology and history of technology.* Cambridge: MIT Press.

Chan, S. 2001. *Complex adaptive systems.* Cambridge: MIT.

Dulac, N., and Nancy Leveson. 2005. Risk analysis of NASA independent technical authority. Cambridge: MIT.

Emes, M., Alan Smith, and Douglas Cowper. 2006. Confronting an identity crisis—How to "brand" systems engineering. *Systems Engineering* 8, no. 2.

Feynman, R. P. 1986. Rogers commission report: Appendix F—Personal observations on the reliability of the Shuttle. NASA.

Forrester, J. 1961. *Industrial dynamics.* Cambridge: Productivity Press.

Hallam, C. R. A. 2001. An overview of systems engineering—The art of managing complexity. Cambridge: MIT.

Heylighen, F., and Cliff Joslyn. 2001. Cybernetics and second-order cybernetics. In *Encyclopedia of Physical Science & Technology* (3rd ed.), ed. R. A. Meyers. New York:Academic Press.

Hughes, A. C., and T.P. Hughes. 2000. *Systems, experts, and computers: The systems approach in management and engineering, WWII and after*. Cambridge:MIT Press.

Hughes, T. P. 1998. *Rescuing Prometheus: Four monumental projects that changed the modern world.* New York: Vintage Books.

Hutchinson, J. 2006. ECE publications manager. J. R. Laracy, ed., email communication September 1.

Jardini, D. 1998. Out of the blue yonder: The transfer of systems thinking from the Pentagon to the great society, 1961-1965. RAND.

Jenkins, G. M., and P.V. Youle. 1971. *Systems engineering: A unifying approach in industry and society.* London: Watts.

Kline, R. R. 2000. The paradox of engineering science. *IEEE Technology and Society Magazine* 19, no. 3: 19-25.

Krenzke, T. 2006. Ant colony optimization for agile motion planning. Cambridge: MIT.

Kroes, P., Maarten Franssen, Ibo van de Poel, and Maarten Ottens. 2004. Engineering systems as hybrid, socio-technical systems. Engineering Systems Symposium, MIT.

Laracy, J. R. 2006. A systems theoretic accident model applied to biodefense. *Defense and Security Analysis* 22, no. 3: 301-310.

Leveson, N. 2002. *System safety engineering: Back to the future*. Cambridge.

Leveson, N. 2004. A new accident model for engineering safer systems. *Safety Science* 42, no. 2.

Mindell, D. A. 2004. Historical perspectives on engineering systems. Engineering Systems Symposium, MIT.

Owens, B. D., Nancy G. Leveson. 2006. A comparative look at MBU hazard analysis techniques. Proceedings of the 9th Annual Military and Aerospace Programmable Logic Devices International Conference (MAPLD).

Scottberg, E. 2006. NASA says shuttle risk overstated; Yet some risk unavoidable. *Popular Mechanics.*

Stamatelatos, M. G. 2002. New thrust for PRA at NASA. NASA.

Sterman, J. 2000. *Business dynamics: Systems thinking for a complex world*. Boston: Irwin McGraw-Hill.

Trist, E. 1981. The evolution of socio-technical systems: A conceptual framework and an action research program. Ontario Quality of Working Life Centre, Toronto.

von Bertalanffy, L. 1969. *General system theory.* New York: George Braziller, Inc.

Weinberg, G. 1975. *An introduction to general systems thinking*. New York: John Wiley.

Wiener, N. 1965. *Cybernetics, second edition: or the Control and communication in the animal and the machine.* Cambridge: MIT Press.

Biography

Joseph Laracy is a PhD student in Engineering Systems and research assistant in the Complex Systems Research Laboratory at MIT. His interests are in system safety and security. He has held engineering positions with Lucent Technologies, Ball Aerospace and Technologies, and Light Source Energy Services. Laracy is a member of the INCOSE, IEEE, and AIAA.

Non-Contacting Interfaces: A Case Study in Modular Spacecraft Design

Sachit Butail, Cornell University
Mason Peck, Cornell University

Abstract

The benefits of modularization, such as reusability and upgradeability, remain underutilized in space. Exploiting these advantages after a spacecraft is launched introduces too much cost and risk in the current paradigm of space-system design. This gap between modular design and its benefits has inspired us to focus attention on interfaces. Many of the obstacles to bridging this gap lie in mechanical attachment of spacecraft subsystems. This paper re-examines the function of mechanical interfaces among spacecraft components and describes a new realization in which flux-pinning superconductors and permanent magnets provide stable, but unpowered, connections among spacecraft components that appear to hover at some distance from one another. Beginning with functional abstraction, this study identifies a new set of unexploited possibilities that add versatility to space-systems architectures. First, the limitations imposed on modularity due to mechanical coupling in spacecraft architecture are identified. These limitations are then used to back out the requirements for a non-contacting interface. With several technologies that enable adhesion without mechanical contact identified, requirements flowdown and functional analysis is performed to quantify technical performance measures for the interface.

Introduction

In the past decade, complex aerospace systems have begun to shift from custom designs towards architectures characterized by standardization and modularization. SMEX, SMEX•Lite, and MightySat are successful examples of modularity in design of space systems. Some commercial space systems, such as those in Boeing's 702 product line, are developed from what can be considered a catalog of subsystems and capabilities. Today, even subsystems are frequently modularized to lower costs in the long run. (Button 2004) proposed a modular power management and distribution (PMAD) system in 2004. While the industry adopts standardization of components as one of the tools to embed modularity in spacecraft design, we argue for focusing on another critical aspect: interfaces.

More than providing a common connection among interacting components, interfaces must be designed to promote reusability and upgradability. Although these benefits are common among terrestrial systems, this trend has only just begun in space. The reasons are the usual ones: cost and risk. Terrestrial systems, because of their accessibility, are much easier to modify during typical operations, costing at worst some unscheduled downtime. Design for changeability (Fricke 2005), a desirable and sought after effect of modularity, is either absent in space systems, or highly limited by the costs and capacity of on-orbit servicing (Davinic 1998). Once a space system is launched, it can rarely be serviced or reconfigured. The Hubble Space Telescope is a rare exception to that rule. One of the major hurdles in bridging this gap is the mechanical coupling between various functional or structural units. The reason is that human or robotic operators are typically required to manipulate mechanical interfaces. However, such resources are virtually non-existent in space.

This paper discusses the benefits of modularity and the role of a non-mechanical interface in spacecraft design. This approach results from exploiting systems engineering techniques. Doing so helps identify a technology that enables non-contacting assembly of systems in space. We rethink the concept of modular spacecraft interfaces, addressing some of the well-known disadvantages of physical coupling, which also opens up new, previously unexploited possibilities for system functionality

We offer a specific implementation as a case study in what matters in interface design. In this case study, rejecting the mechanical approach not only promises improved system operation, the interface even becomes a means for engineers to realize new types of spacecraft. Some of the relevant trade studies are summarized. We report the results from a demonstration of this interface: preliminary tests with two-subsystem modules on three and five degree-of-freedom testbeds successfully verify the stability of a candidate arrangement of magnets and superconductors. The verification includes assessments of apparent stiffness, versatility in separation, contact-free docking/berthing, and reconfigurability of the modular design.

Non-Contacting Interface—Motivation

There are several reasons why a non-contacting interface fares better than one that requires physical, though detachable, coupling. Low wear and tear and the resulting high reliability are the most obvious. In addition, electrostatic discharge upon contact is far less likely; the possibility of sticky or otherwise failed mechanical interfaces need not be accommodated by large actuator forces; special handling techniques for bolting together hardware in space are irrelevant (Roberts 1998).

A non-contacting interface promises all the benefits of an n-modular system such as versatility, robustness, and low cost (Duff 2001). The interface can be controlled actively—allowing articulation among components—and therefore, allows spacecraft to be reshaped. For example, modules fitted with optically reflective surfaces can morph into a mirror with variable focus, allowing near or far objects to be imaged with a single spacecraft. Finally, beyond the inherent reduction in cost from a modular design (Enright 1998), a non-contacting interface allows less-expensive repair, greater fault tolerance, and the prospect of an incremental build.

Starting with several trade studies, a team at Cornell University has successfully built and demonstrated a mechanically stiff arrangement of non-contacting spacecraft modules by using superconductors and magnets. This demonstration system depends on superconductive flux pinning to achieve a non-contacting mechanical coupling, and thus a reconfigurable, modular spacecraft. Flux pinning is a term that refers to a special behaviour of certain Type II superconductors, such as Yttrium Barium Copper Oxide (YBCO). This material does not simply reject magnetic fields from its interior, as do most superconductors. Rather, if the magnetic flux is dense enough, the superconductor holds onto flux lines, constraining them in five degrees of freedom (two rotations and three translations). With multiple superconductor / magnet pairs across the interface, all degrees of freedom can be pinned: in other words, one module is constrained to move with its neighbor. This interface also turns out to offer significant structural damping, which dissipates the energy associated with vibration, helping to eliminate vibrations across the spaces structure and enhancing the settling time of reconfiguration maneuvers.

Such a system makes sense in space. At the low temperatures available in space (except in full sun), little power, if any, must be devoted to maintaining the low temperatures (about 80K) required for superconductivity. In a microgravity environment, disturbance forces and torques that would tend to dislodge the non-contacting modules are negligible. Furthermore, with considerable spatial extent available, both n-modular (n similar modules) and functionally modular designs are conceivable. In fact, a spacecraft based on this non-contacting architecture is both physically and functionally flexible. One is inclined to think of modules as LEGO® blocks that could be arranged in any fashion. The non-contacting interface will literally ease the detachment and reattachment to achieve a given configuration. The hysteretic damping associated with flux pinning is not Coulomb friction; there is no stiction to overcome in releasing one module from another. Of particular interest is the fact that warming a superconductor past its transition point, moving the magnetic field, and re-cooling the superconductor, allows a new equilibrium configuration to be "recorded" by the superconductor. Thus, maintaining a new configuration also requires little if any power. This functionality compares favorably to other proposed approaches (Miller 2002), in which power must be devoted to actively stabilize magnetic fields.

Modularity in Spacecraft Design: A Comparison

Here we list some of the salient principles of modularity and assess how a non-contacting interface demonstrates or fails to exhibit them:

a. A modular architecture has well defined, standardized, and decoupled interfaces (Holmqvist 2003).
b. A modular design has encapsulation with simple interfaces.
c. Modular systems have loose coupling and strong cohesion; reuse of elements and modules is possible.
d. A modular architecture entails a complete set of design rules (Baldwin and Clark 1999).

The superconducting interface proposed here will have the capability to clamp surfaces at a distance. Like any other mechanical interface, say a linear spring with a constant stiffness, the non-contacting surfaces can also be evaluated in terms of simple technical performance measures like stiffness and damping. The degree of encapsulation depends on the interdependence within units: besides functionally committing to this principle, a non-contacting interface features physically encapsulated units as well. As discussed below, the ease of coupling provides a better platform for reuse of modules.

Therefore, a non-contacting interface remains consistent with the principles of modularity. But there are in fact more benefits of modularity that may not be immediately apparent. Table 1 compares one of the most modular systems, a personal computer, to a spacecraft to highlight simple differences and show where there is room for improvement.

Table 1. Extent of Modularity

Feature	Personal Computers	Spacecraft
Plug and play	Almost every part of a computer comes as a plug-and-play device. A feature can be added or removed any time to the system.	While plug-and-play capability may be possible on ground, in space it is highly limited by cost and complexity
Hardware System upgrade & repair	Hardware system upgrade and repair is possible at any time, and is as easy as replacing a part. Redundant system is discarded physically.	Hardware system upgrade and repair requires on-orbit servicing, a costly procedure. Functional servicing increases mass penalty.
Scalability	As a processing unit, computers can be clustered to perform more complex tasks with increased capacity.	A satellite constellation is the closest example to a clustered system of spacecraft operating at higher capacity. However, such space-system architectures amount to physically arranging them in a desired configuration, which again, brings us back to the limitations (cost and risk) of in-orbit assembly and maintenance.
Assembly	Computers can be assembled at ease on ground	In-orbit assembly faces at least the same challenges as docking in space.

The above comparison reveals a stark contrast--not in the extent of modularization, but in the advantage of modularity in spacecraft design. System upgrade, scalability, and assembly are some of the major cost savings that result if one adopts modularity in design. But for a spacecraft, these features are very expensive and risky. On-orbit servicing promises to solve part of the problem. A non-contacting interface, on the other hand, approaches from a different direction, exploiting modularity by a sudden design rationalization (Baldwin and Clark 1999).

Systems Approach

Interface Requirements

Over the past two years, a small team at Cornell University has performed tests and trade studies to arrive at favorable technologies for a non-contacting interface. Systems Engineering techniques, such as requirements analysis, functional-flow diagrams, trade studies, and structural modeling were used to make a seemingly uncertain goal attainable. The proposed interface is a subsystem that is meant to reside on all modules of a space system to be assembled in orbit. What follows is a discussion of some of the top-level requirements that drive the functional analysis and subsequent trade studies for design.

We will begin by defining a few terms:

- *Non-contacting*: Modules that interact by action at a distance. All phenomena that occur without any direct physical contact between bodies fall into this category. Examples are field forces such as magnetic, electrostatic and gravitational.
- *Interface*: According to (Kossiakoff 2003) interfaces are "a critical systems engineering concern, which effect interactions between components; interfaces include elements that connect, isolate, or convert interactions" The interaction media include mechanical, electrical, hydraulic or data. A simple example of an interface is a three-pin plug and socket used to connect an electrical appliance mechanically to power mains.
- *Module(s)*: Modules in this context are any two bodies that are physically detached. They may represent functionally separate subsystems or small systems of the same kind performing a collective function. It is proposed that a non-contacting interface is present between two interacting modules in space.

We have defined the following top-level requirements for a demonstration system consisting of two modules with a mass budget of 400 grams each.

- *Physical contact*: The interface shall require no contact by mechanical means.
- *Stability*: The relative position and orientation of the two modules shall tend toward a defined equilibrium in finite time.
- *DOFs*: The interface shall demonstrate a stable arrangement of modules in at least three degrees of freedom. This requirement is progressively extended to six degrees of freedom.

The aim is to design a stable system countering typical disturbances for an orbiting spacecraft. By enabling angular and translational motion between the modules, the system as a whole dissipates energy and remains intact. It is important that both these relative motions are permitted

without any use of power so that the non-contacting interface represents no significant additional drain on this valuable resource.

Derived from these high-level requirements are the following, which lead to certain architectural decisions. Failures that prevent other subsystems from recovering autonomously within 15 minutes are sufficiently severe that retaining control of the modules for performance reasons makes little sense.

To preclude system deformation at a range of orbits due to orbital perturbations, a stiffness value of more than 7.5 N/m is required. The requirement stems from a target structural frequency of 1Hz for modules weighing 400 g. Heavier modules will demand higher stiffness values.

A separation distance of 5mm flows from two higher-level requirements. First, it represents some margin for perturbations while preventing collisions and interference between bus and payload electronics and second, it also provides some degree of articulation of the system—i.e. motion of one body relative to its neighbor. Articulation enables the system's morphology to change in response to the environment (e.g. to mitigate environmental torques in space) or to changing mission requirements (e.g. a change in focal length of a sparse-aperture paraboloid mirror). Active feedback control would be used for this purpose, and linear (small-angle) motions will simplify the controls-algorithm implementation.

The settling time must be as low as possible so that mating dynamics do not unduly interfere with mission operations. A high damping ratio can also eliminate the risk of unwanted contact due to persistent oscillations. Backtracking from one of the candidate technologies allows us to establish a requirement of 10 seconds for the proposed interface.

- *Unpowered Operation*: The interface shall maintain stability without power for at least fifteen minutes.
- *Settling time:* The modules, upon mating, shall settle into a stable arrangement within 10 seconds.
- *Stiffness*: The stiffness of non-contacting interface shall be at least 7.5 N/m.
- *Separation distance*: The distance between the modules shall be at least 5mm.
- *Active Reconfiguration*: The interface shall permit variation in stiffness by application of power to electromagnetic actuators.
- *Non-contacting Power transfer*: The interface shall permit transfer of power between the modules.
- *Non-contacting data transfer*: The interface shall permit transfer of data between the modules.
- *Non-interference with other spacecraft functions*: The interface shall not interfere with the spacecraft payload operations.

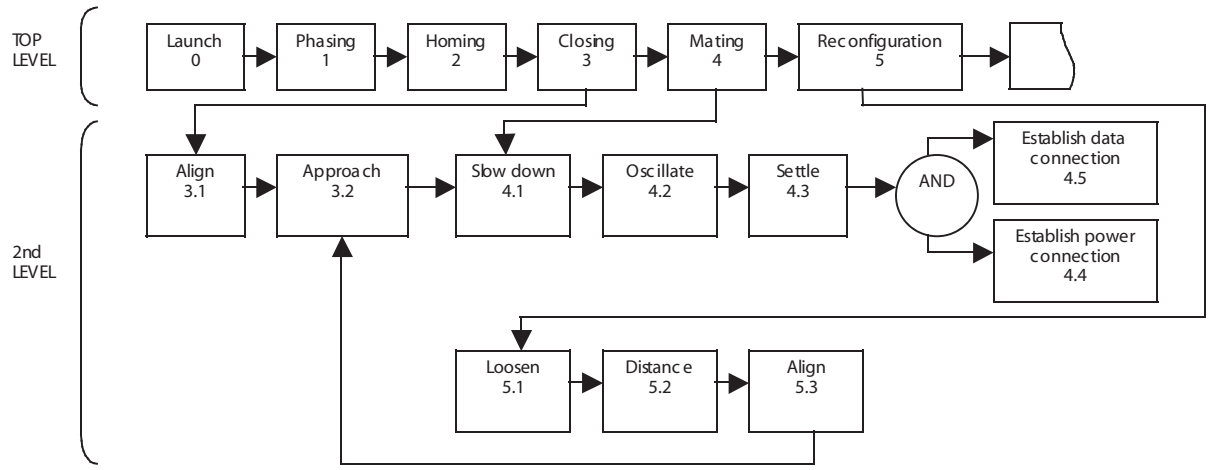
These requirements have been flowed down into structural specifications.

Functional Flow Block Diagram

Here we describe the functional analysis that supports this design approach. It highlights the principle that loose constraints in function allocation allow flexibility in technology selection, and creation of quantifiable sub-requirements.

Figure 1 is a Functional Flow Block Diagram (FFBD) that describes the mating stage of two modules, say a target and chaser, that comprise a rendezvous mission. In this case, far-range rendezvous, or homing, starts at a range of few kilometers and ends when the two modules are separated by a few meters. The close-range rendezvous, or closing, then requires the modules to be aligned in a favorable configuration. Once the modules are aligned, the probability of a collision is much lower than in traditional docking procedures because physics establishes

Figure 1. Functional Flow Block Diagram for In-orbit Rendezvous Using Non-contacting Interfaces Between Modules



the desired equilibrium rather than potentially faulty actuators or control algorithms. The mating operation, which has also been demonstrated in the laboratory, can begin with the modules a few centimeters apart. An arrangement of permanent magnets creates an artificial potential well which pulls the chasing module into the desired position.

Trade Studies

Trade studies were carried out to identify a suitable realization of the functionality. The resulting non-contacting interface comprises three main components:

1. The top-level requirements for mechanical interface are few, and yet drive the design. The non-contacting mechanical coupling is the by far the most important feature of the proposed interface. The aim for this demonstration system is to prove the concept but not preclude more general uses of the architecture in a follow-on project. Some of the admittedly ambiguous but key criteria that flow from this principle and drive the design of the demonstration system are as follows:

- Traceability to space
- Appreciable separation distance
- Minimal interference with spacecraft bus and payload
- Large basin of attraction allowing robustness
- Power required

The demonstration system was based on a three degree-of-freedom testbed, a planar arrangement in which one translation and two rotations were constrained. Although the stability of the demonstration system is based on this 3 DOF design, that the approach is meant to be sufficiently generic for a follow-on 6 DOF system.

There are several types of forces that do not require physical contact that are candidates for the mechanical component: magnetic forces, electrostatic forces, and gravity. We provide an overview of these options here to defend the choice of flux pinning as the clear preference from a system perspective. All are inverse-square forces (i.e. the force varies with the inverse of the square of the distance between bodies). Earnshaw's Theorem states that no divergenceless force (such as these) can result in a stable static equilibrium. So, no combination of fixed magnets and electric charges can levitate an object stably in gravity. Some ways around the theorem take advantage of Earnshaw's assumptions. Based on the criteria for trade study, the following options for contact-free interfaces were evaluated:

- Quantum effects, according to which electromagnetic intermolecular forces cause any material we perceive as "touching" another to be displaced by some nearly imperceptible distance from it: This quantum distance is not useful for our purposes.
- Feedback control, which moves the magnets (or temporally varies their fields): This approach would provide stable action at a distance, and has been proposed for formation flying (Miller 2002). However, it requires power, interacts detrimentally with spacecraft electronics, induces unwanted, attitude-perturbing torques due to surrounding fields (such as the geomagnetic field), and introduces the very real risk that a temporary loss of power or a software failure may cause the assembly to lose structural integrity.
- Oscillating and moving magnets, whose quasi-passive, periodic motion creates relative equilibria (in the Hamiltonian sense): An entertaining example of this behavior is popular Levitron toy (Gov et al. 1999, 2000), in which one spinning magnet levitates several inches above another. Because it depends on bound angular momentum, this principle is not particularly useful for spacecraft, where angular momentum is carefully managed for attitude control. Also, outside the relatively small stable region, the levitated magnet is unstable and exhibits unwelcome, energetic dynamics.
- Diamagnetism, the property of many high-temperature superconductors (Type I and many of Type II) and some room-temperature solids such as pyrolytic graphite (Simon 2001) that magnetize in the direction opposite to a magnetic field in which they are placed: Maglev trains exploit this property, using the Meissner-Oschsenfeld expulsion of a magnetic field for stable levitation. Our recent experimentation with room-temperature diamagnetic materials and relatively common rare-earth permanent magnets has convinced us that the separation distances are too small for any of the many advantages a non-contacting interface ought to offer. Further, the Meissner effect associated with these superconductors offers very low stiffness in many degrees of freedom and a small basin of attraction and is therefore is not as effective as the flux-pinning approach we favor.
- Flux pinning, another property of Type II superconductors, notably the YBCO variety. In these materials, vortex-like supercurrent structures in the material create paths for the flux lines. When the external sources of these flux lines move, however, these supercurrent vortices resist motion or are "pinned" in the superconducting material. This so-called flux-pinning is the source of stable levitation in these materials (Brandt 1990; Moon 1994). These forces are surprisingly strong and can be engineered to fix all six degrees of freedom of one rigid body relative to another. What is required is a DC magnetic field, such as from a permanent magnet, and the appropriate Type II material. When these substances are within a certain distance and orientation of one another, the two find a stable, static equilibrium.

All of the requirements are satisfactorily met using flux pinning superconductors and permanent magnets. None of the other approaches offer even similar performance. For a single HTS-permanent magnet pair, at a distance of approx. 6mm and magnetic moment vector normal to the surface of the HTS, a simple experiment revealed stiffness values ranging from 17 N/m to as high as 130 N/m on various degrees of freedom with the highest value usually being in the direction of maximum flux change. This stiffness depends on the orientation and distance between the permanent magnet and superconductor, the magnetic field, and the material of the superconductor (Moon 1994). There is also a sixth degree of freedom along the magnetic axis which exhibits no damping. Any angular disturbance along this direction makes a magnet spin continuously. Our tests showed a high amount of damping along all other degrees of freedom.

To be able to confidently accept or reject certain alternatives, we reconcile them with critical requirements discussed in the Systems Approach section. The following decision matrix describes our evaluation of flux pinning for a non-contacting interface. Because all of the criteria listed below are important for realization of the said interface and each criterion has equal significance, the weight is also the same.

The rating scale options are: +1 = satisfies the criterion; 0 = No basis for judgment; -1 = does not satisfy the criterion.

2. The Electrical Power System (EPS) can be either a distributed system or a single subsystem, i.e., the power system can either locally reside (with solar panels and energy storage device) on each module, or it can be centrally placed and transfer power wirelessly to remaining parts.

Trade studies were conducted to find the best possible technology for non-contacting power transfer across modules. The criteria used to evaluate various technologies were efficiency, mass, size, and extent of interference with the mechanical component of the non-contacting interface. A combination of mono-crystalline photovoltaic cells and infra-red light emitting diodes (LED) with matching wavelength was found most efficient with a net approximate value of 5% in our tests. However, considerably better performance is likely.

3. Data can be wirelessly transmitted by using the best available technology subject to following criteria: mass, size, interference with non-contacting forces, and extent of interference with wireless power transfer. An increased responsibility of the Command and Data handling sub-system will be to communicate relative positions and coordinate reconfiguration maneuvers.

Potential Well

The equilibria of these modules are within potential wells with basins of attraction that can be quite large. Using only small magnets and superconductors, we have demonstrated assembly of components from within many centimeters of each other to reside stably at a distance of about 1 cm. The creation of a passive potential well using a simple arrangement of permanent magnets and superconductors gives this kind of interface a distinct advantage over mechanically mating docking systems. One of the mechanical interface problems during on-orbit servicing is precise alignment during mating operations (Moynahan 2001). A system that falls into a stable potential well all by itself is free from such concerns.

Table 2. Decision Matrix

Alternatives → Criteria ↓	Feedback Control	Moving Magnets	Diamagnetism	Flux Pinning
Unpowered Operation	-1	+1	+1	+1
Less settling time	+1	-1	+1	+1
High stiffness	+1	0	-1	+1
Separation distance more than 5mm	+1	+1	-1	+1
Ability to actively reconfigure	+1	-1	0	+1
Large region of stability	+1	-1	-1	+1
Total	+4	-1	-1	+6

Figure 2. Examples of Unstable Potential

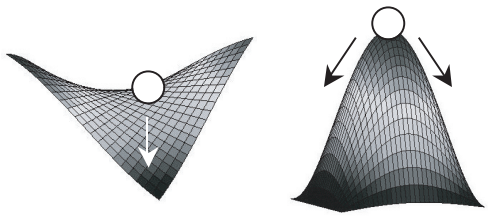


Figure 3. Potential Well for a Current Loop

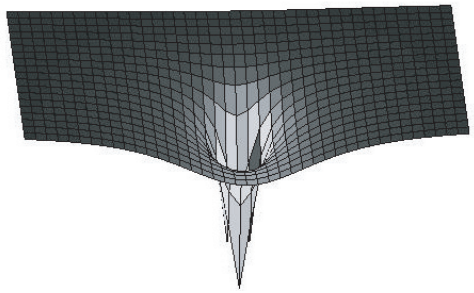


Table 3 describes the impact of a non-contacting interface in terms of the scenarios in Table 1: Extent of Modularity. It shows that a non-contacting interface helps achieve modularity in implementation as well.

A spacecraft with non-contacting interface can exploit many advantages of modularity that are available to other complex systems on earth. Several possible scenarios emerge:

- Small modules launched as payloads find each other to form a larger system
- The self-assembly requires no appreciable power, and yet all parts are stably, stiffly positioned without contact
- A spacecraft is upgraded to higher capacity by sending tiny modules as part of excess launch capacity.
- An inventory of modules is maintained in space that can be used to replace or add to existing spacecraft.
- A faulty part on the spacecraft is replaced physically in a fraction of the cost of traditional on-orbit servicing.
- A spacecraft comprising tiny modules is able to morph itself into wide range of shapes and alignments to fulfil a given task.
- A large robotic arm, not articulated by traditional rotational and prismatic joints but rather by the component-to-component control.

Conclusion

Flux pinning in superconductors is a relatively unfamiliar concept in engineering. It is entirely unknown in the context of modular interfaces. While the physics of this phenomenon is still the subject of fundamental research, one of our main goals is to be able to model the interaction between a superconductor and magnetic field subject to change in parameters such as distance, size, mass, and magnetic strength. Preliminary tests with two modules on a three and five degree of freedom testbed have successfully confirmed the ability for such modules to achieve a desired, stable arrangement. An appreciable stiffness is also attained between the superconductor and the permanent magnet once pinned in position. Although reconfiguration using electromagnetic forces is part of ongoing research on electromagnetic formation flight (Miller 2002) the broader goal of less costly and simpler in-orbit assembly, reconfiguration, and on-orbit servicing can be achieved by using a naturally occurring effect such as one proposed in this paper.

Acknowledgments

This work was supported in part by the NASA Institute for Advanced Concepts.

Table 3. Modularity With a Non-Contacting Interface Applications

Feature	Non-contacting interface
Plug and play	With a relatively simple and less complex docking interface, subsystems can attach and start to function in space as well.
Hardware system upgrade and repair	For better upgradeability, modules can have a non-contacting interface available for similar subsystems. Autonomous physical servicing becomes possible.
Scalability	Many spacecraft with a non-contacting interface can combine and reconfigure to increase capacity/ functionality.
Assembly	Spacecraft parts or modules can be launched separately, or, by using the excess launch capacity on launch vehicles

References

Baldwin, C.Y. and K.B. Clark. *Design rules. Vol. 1: The power of modularity.* Cambridge, MA: MIT Press.

Brandt, E. 1990. Rigid levitation and suspension of high-temperature superconductors by magnets."*American Journal of Physics, 58,* no. 1: 43-49.

Button, R. and J. Soeder. 2004, August. Future concepts for modular, intelligent aerospace power systems. International Energy Conversion Engineering Conference, Providence, Rhode Island.

Davinic, N., A. Arkus, and S. Chappie. 1998. Cost-benefit analysis of on-orbit satellite servicing. *Journal of Reducing Space Mission Cost, 1*, no. 1: 27-52.

Duff, D., M. Yim, and K. Roufas. 2001. Evolution of Polybot: A modular reconfigurable robot. Proceedings of COE/Super-Mechano-Systems Workshop.

Earnshaw, W. 1842. On the nature of the molecular forces that regulate the constitution of the luminferous ether. *Trans. Camb. Phil. Soc., 7*: 97-112.

Enright, J., C. Jilla, and D. Miller. 1998. Modularity and spacecraft cost. *Journal of Reducing Space Mission Cost, 1:* 133-158.

Fricke, E. and A. P. Schulz. 2005. Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. *Systems Engineering 8,* no. 4: 342-359.

Gov, S., S. Shtrikman, and H. Thomas. 1999. On the dynamic stability of the Hovering Magnetic Top. *Physica D 126:* 214-224.

Gov, S., S. Shtrikman, and H. Thomas. 2000, April. 1D toy model for magnetic trapping. *American Journal of Physics, 68,* no. 4: 334-343.

Holmqvist, Tobias K.P. and M.L. Persson. 2003. Analysis and improvement of product modularization methods: Their ability to deal with complex products. *Systems Engineering, 6,* no. 3: 195-209.

Kossiakoff, A. and W. Sweet. 2003. *Systems engineering principles and practice.* Wiley and Sons, pp. 45-46.

Miller, D.W., R.J. Sedwick, E.M.C. Kong, and S. Schweighart. 2002. Electromagnetic formation flight for sparse aperture telescopes. *IEEE Aerospace Conference Proceedings, 2*.

Moon, F. 1994. *Superconducting levitation: Applications to bearings and magnetic transportation*. New York: Wiley.

Moynahan, S.A., III and S. Touhy. 2001. Development of a modular on-orbit serviceable satellite architecture. *20th Conference on Digital Avionics Systems*, 2: 14-18.

Roberts, B. 1998, May. Manufacturing and testing requirements for a reversible hand socket wrench using three-dimensional rollers. NASA/CR-1998-206849.

Simon M.D., L.O. Heflinger, and A.K. Geim. 2001. Diamagnetically stabilized magnet levitation. *American Journal of Physics, 69,* no. 6: 702-713.

LEGO is a registered trademark of LEGO Company

Biographies

Sachit Butail did his Bachelors in Mechanical Engineering from Delhi College of Engineering, Delhi University, India (2000) and Masters in Systems Engineering from Cornell University (2005). He has worked as an Assistant Systems Engineer and Analyst at Tata Consultancy Services. He has been actively involved in Professor Peck's team for his research on non-contacting interfaces for the last two years and is currently a Research Assistant in the Space Systems Design Studio at Cornell University.

Mason Peck, PhD, is an assistant professor in Cornell's Sibley School of Mechanical and Aerospace Engineering and in Cornell's Systems Engineering Program. Since 1993 he has held various systems-engineering and analysis positions in the aerospace industry, most recently Principal Fellow for Honeywell Defense and Space Electronics. He holds a Ph.D. and an M.S. in Aerospace Engineering from UCLA, an M.A. in English Literature from the University of Chicago, and a B.S. and B.A. from the University of Texas at Austin.

Recent Research on the Reliability Analysis Methods for Mobile Ad-hoc Networks

Jason L. Cook, Picatinny Arsenal
Jose Emmanuel Ramirez-Marquez, Stevens Institute of Technology

Abstract

The Mobile Ad-hoc Wireless Network (MAWN) is a network schema that does not require the infrastructure items (e.g., cellular towers) of typical networks. The flexibility this offers has led to the proliferation of this network schema. However, this unique attribute of ad-hoc networking also violates the base assumptions on which existing network reliability methods are founded. That is, that the configuration of the network is known *a priori*. This paper will describe methods being developed to fill the void left by existing techniques. This paper will also describe the utilization of these methods in a systems engineering construct.

Introduction

The reliability of a Mobile Ad-hoc Wireless Network (MAWN) is paramount in its prevailing applications; such as for DoD and First Responder networks. However, to achieve system reliability the system engineer must first be able to define and measure this metric. The challenge is that the MAWN does not conform to the basic assumption on which existing network reliability methods are founded. So, the existing reliability analysis methods are inappropriate and incapable of measuring the reliability of the MAWN.

For infrastructure-based networks the configuration of a network is known and mostly constant. In other words, the structure of the network in terms of component connectivity is known *a priori*. Accordingly, the component-wise relationship to reliability can be depicted graphically with methods such as reliability block diagrams (RBD) and fault tree analysis (FTA). Similarly, it is possible to rigorously develop a closed form expression to express this relationship mathematically or for more complex systems it is possible to develop a mathematical approximation based on cut-sets and other techniques as detailed in Ebeling (1997, 5). However, due to the MAWN's dynamic formation and reformation, the reliability block diagram or reliability expression that represents the system changes with time.

This paper will define reliability metrics for the MAWN propose new methods that account for these features so that these metrics may be calculated.

Literature Review

Research in the field of network reliability generally focuses on communication between members of the network referred to as nodes or terminals. The prevailing metrics used then is the probability of a successful connection between two or more terminals. This metric is *k*-terminal reliability (kTR). Two popular modifications of this metric are two-terminal reliability (2TR) and all-terminal reliability (ATR). Respectively, these are simply when *k* is equal to two or to the number of terminals in the network.

These metrics may be determined via rigorous development of the reliability expression or approximated via cut set techniques but many researchers have developed other methods to improve upon this type of analysis. As an example, Rocco and Muselli (2004) developed network reliability methods using machine learning techniques to account for 2TR when node and link capacity are incorporated into the model. Further, Ramirez-Marquez and Coit (2004) proposed a heuristic method to address both multi-state and capacitated network reliability.

There has been a relatively few attempts in analyzing reliability of cellular and other infrastructure based wireless networks, one such contribution comes from Chen and Lyu (2005). These authors illustrate the process of handoff in a mobile cellular network; the transition of a mobile cellular phone's linkage from one cell tower to another. These transitions happen as a cellular user moves from the coverage area of one tower to the area covered by the other. Markov models were used to represent this configuration change and expressed network reliability as a function of the reliability of each node active in the configuration and the percentage of time that each configuration exists. However, this method is not directly applicable to a MAWN the major assumption is that the failure of any active node in the message's route results in failure, and as such, the reliability model is always represented by a configuration in series. This is not the case in a MAWN, because redundant paths may exist between source and destination.

So, despite the published methods that apply to wireless networks, they still do not address the need for methods for MAWN. Specifically, they do not address the unique characteristics that make the MAWN a valuable network

type. These wireless methods still focus on a network configuration that is relatively constant and stable.

The modeling of node mobility is also important for MAWN reliability and some of the existing models and methods will be applied within this paper. A comprehensive survey of current mobility models describing the mobility patterns of each model and comparing several metrics relevant to MAWN performance was presented by Camp et al (2002). One of the mobility models is the Random Waypoint Mobility Model (RWMM).

The RWMM has been widely applied for modeling the mobility of a MAWN. The model describes the motion of a Mobile Node (MN) that travels in a randomly selected direction, at a randomly selected speed, for a certain amount of time. The MN then selects a new direction and speed. This model is run within a simulation boundary representing the expected coverage of the MAWN. Once a MN reaches this specified boundary, it changes direction and moves back into the area.

These models have been used to evaluate different network protocols for implementation in MAWN but none have not previously been utilized for reliability evaluation or analysis in the manner that will be demonstrated within this paper.

It is also critical to consider the application of the MAWN when developing these methods. The DoD application will be applied throughout this paper by utilizing node and network parameters that are representative of DoD networks. The Department of Defense (DoD) employs the MAWN to enable tactical communications. Freebersyser and Leiner (2001) reported on several MAWN developed for military use—the DARPA Packet Radio Network (developed in 1972) and the 1997 Task Force XXI Advanced Warfighting Experiment are two of these.

Technical Content

General Problem Formulation

Let **G** = (**N**,**L**) represent a MAWN where **N** the set of nodes and **L** is a matrix that represents the links between the nodes. The elements of **N** shall be n_i for i=1,2..n where n is the number of nodes in **N**. Then, the nodes' operational status at time, t, shall be $n_i(t)$ where $n_i(t) = 1$ if node i is operational, else $n_i(t) = 0$. The elements of **L** shall represent the wireless links between nodes i and j as $l_{ij}(t)$ for every combination of arcs i,j = 1,2,..,n. Let $l_{ij}(t) = 1$ if the link between the nodes exists else let $l_{ij}(t) = 0$. The reliability associated with each node of network is represented by $r_i(t)$ and let $v_{ij}(t)$ represent the probability associated with $l_{ij}(t)$ existing (i.e. $v_{ij}(t)$=P($l_{ij}(t)$=1)). The model chosen to imitate node mobility is RWMM and thus, the resulting distribution is uniform; as shown by (Camp et. al 2002) resulting in $v_{ij}(t)=\lambda\ \forall\ i,j$ where $i \neq j$ and at all points in time. Let **C** define the set of possible network configurations. In a MAWN the existence of a link in **L** is probabilistic and each combination of existing and non-existant links represents a different potential network configuration. The number of potential network configurations is given by:

$$|\mathbf{C}| = 2^{n*(n-1)/2} \tag{1}$$

$2TR\alpha_k$ defines the two-terminal reliability of configuration α_k, k = 1, 2…|**C**|. Finally, let $2TR_m$ be the probability that a communication path exists between the source and destination nodes.

Method for 2TR$_m$

The network is defined an each possible permutation of link states provides insight into the number of configurations the MAWN may take on over time. The probability of each configuration existing may be determined as a function of the link probability of existence, λ, the number of linked node pairs, η_l, and the number of unlinked pairs, η_u, in the configuration. The probability associated with each possible configuration is given by:

$$P(\alpha_k = 1) = \lambda^{\eta_l}(1-\lambda)^{\eta_u} \tag{2}$$

Finally, the $2TR_m$ may be obtained as a weighted average of the probability of existence for each configuration and the associated reliability. The result is the two-terminal reliability for the MAWN. Mathematically, this is expressed as:

$$2TR_m = \sum_{k=1}^{|\mathbf{C}|} 2TR\alpha_k * P(\alpha_k = 1)$$
$$2TR_m = E[2TR\alpha_k] \tag{3}$$

The method to perform this calculation follows:

1. Define n, r_i, and λ
2. Enumerate all possible configurations of **G**(**N**,**L**) and stack them in set **C**.
3. Determine P(α_k=1).
4. For k=1,..|**C**|, obtain $2TR\alpha_k$ based on r_i considering links in the configuration to have perfect reliability.
5. Calculate $2TR_m$.

Illustrative Example

A three node network will be used to further illustrate this method. Consider, a network where n = 3, r_i = 0.9 and λ = 0.7.

The three node network has eight potential configurations, shown in Figure 1.

After enumeration of each configuration, the reliability and probability of existenance for each configuration is calculated and subsequently, the $2TR_m$ is calculated, results are shown in Table 1.

Figure 1. Network Configurations

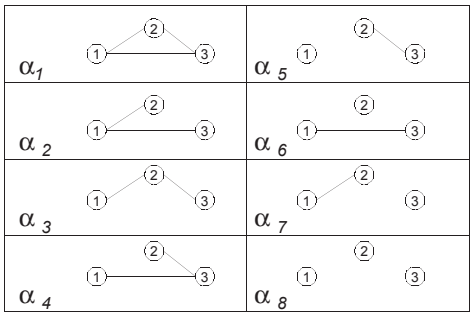


Table 1. $2TR_m$ for 3 Node MAWN

α_i	l_{12}	l_{13}	l_{23}	P(α_i = 1)	$2TR\alpha_i$
1	1	1	1	0.34	0.81
2	1	1	0	0.15	0.81
3	1	0	1	0.15	0.73
4	0	1	1	0.15	0.81
5	0	0	1	0.06	0.00
6	0	1	0	0.06	0.81
7	1	0	0	0.06	0.00
8	0	0	0	0.03	0.00
					$2TR_m$ = 0.6742

After enumeration of each configuration, the reliability and probability of existenance for each configuration is calculated and subsequently, the $2TR_m$ is calculated, results are shown in Table 1.

The enumeration method provides an exact solution, yet the method becomes computationally expensive for the analysis of large networks. Thus, a simulation technique has been developed to develop an accurate approximation with less computing burden, as a means to estimate 2TRm. This MC simulation approach includes simulating the operational state of the nodes and links as described in pseudo-code below. The nodes are simulated first and then the links, acknowledging that no failed node can be linked.

Procedure to Simulate Node Status
for i=1,2…n
test ← select random
if *test* ≤ r_i then n_i =1
else n_i=0
N← n_i

Procedure to Simulate Link Status
for i=1,2…n
for j= i+1,2…n
test ← select random number
if (*test* ≤ $\lambda \cap n_i$ =1 $\cap$ n_j =1) then l_{ij}=1
else; l_{ij}=0
$l_{ji} = l_{ij}$
L← l_{ij} and l_{ji}

After simulating node and link status, the resulting link and node states are compared against the success criteria; a path between source and destination exists. This is done by analyzing the link configuration matrix, **L**, to determine all the nodes that have a path to and from the source node. A connectivity vector is defined with a length n as Λ. Let Λ_i = 1 if node i is connected to the source node; therefore if a path exists from source to destination node which is numbered n then Λ_n = 1. Λ is then populated by performing a breadth first search on **L** to see if any combination of l_{ij} creates a path from between source and destination node.

These procedures are used to generate the following MAWN simulation approach, where Q is the number of runs in the simulation.

Calculation of $2\hat{T}R_m$
for q=1,2…Q
Simulate Network → **L**(q)
Find Connectivity → $\Lambda_n(q)$

$$2\hat{T}R_m = \frac{\sum_{q=1}^{Q} \ddot{E}_n(q)}{Q} \tag{4}$$

The results of the simulation as compared to the complete enumeration procedure are presented in Table 2.

Table 2. Simulation Results

# Nodes	$2TR_m$ Results: Complete Enumeration	Simulation	Relative Simulation Error
3	0.6742	0.6770	0.42%
4	0.7461	0.7498	0.50%
5	0.7837	0.7882	0.57%
6	0.8001	0.8083	1.02%

This method may be utilized to understand the relationship between system (network) parameters and their impact on reliability. To demonstrate this, the relationship between n and λ is investigated. In comparing Figures 2 and 3, it is evident that a network with more nodes can achieve a higher level of reliability with a lower probability of link existence.

Additional utilizations of this method and more examples and results have been developed. For the additional techniques, more results, and full mathematical formulation and derivations refer to Cook and Ramirez-Marquez (2006).

Mobility and Reliability Model

The same notation and metrics are applied to gain even

Figure 2. $2TR_m$ vs. n

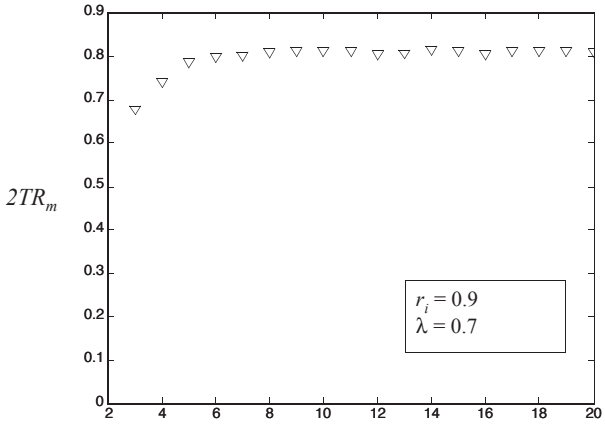
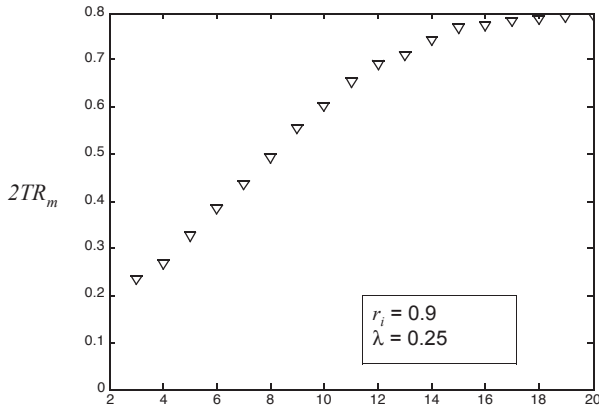


Figure 3. $2TR_m$ vs. n



further insight into the reliability of a MAWN. Here, we remove the assumption that probability of link existence is known by embedding a mobility model and determining link existence from the relative location of nodes with respect to each other and their transmission distance. For this purpose, define node separation distance, $d_{i,j}$, as the distance between node *i* and *j* for all nodes in G. Then, let a link exist between nodes *i* and *j* if their separation distance, d_{ij}, is not greater than their transmit/receive range, t_{ij}.

Because the nodes' position changes due to their mobility, then the connectivity matrix **L** representing the network configuration also changes with time and becomes **L**(*t*). The RWMM described by (Camp *et al* 1997) is used as the mobility model. The mobility of a given node is defined by the linear velocity, v_i, and heading, φ_i. Simulation of these values allows for the determination of each nodes position as a function of time, $d_{ij}(t)$.

Again, define the connectivity of the network, Λ, where $\Lambda_i(t)$ is the connectivity state of the i^{th} node at time *t* with respect to the source node. That is, $\Lambda_i(t) = 1$ if the i^{th} node has a path to the source node at time *t*, else $\Lambda_i(t) = 0$. Then the two-terminal reliability (2TR) of the MAWN, notated at $2TR_m$, is given by the following equation:

$$2TR_m(t) = \mathrm{P}(\Lambda_i(t) = 1) \tag{5}$$

Introducing the time element to the problem, also allows us more fidelity on node reliability. Allowing for the consideration of reliability degradation overtime. The Weibul distribution is selected to describe node reliability

$$r_i(t) = e^{(-t/\theta)^{\beta}} \tag{6}$$

The calculation of $2TR_m$ is done in the same fashion as the simulation described previously. Several iterations of the simulation are run and the results of each define an instance of **L**, which is analyzed to determine **Λ**. The results of each run are then collected and reliability is determined by:

$$2\hat{T}R_m(t) = \frac{\sum_{run=1}^{Q} \Lambda_i(q,t)}{Q} \tag{7}$$

kTR_m may also be determined by analyzing **Λ.** For each run of the simulation and determining from the results the probability that more than *k* terminals are connected.

$$kTR_m(t) = \mathrm{P}\{\sum_{i=1}^{n} \ddot{E}_i(t) \geq k\} \tag{8}$$

Illustrative Example

A network with the following parameters is analyzed; n = 18 , $r_{i,j}$ = 3 miles $\forall$ *i,j* θ = 1000 β = 1.5 , v_{max} = 6 mph and v_{min} = 3 mph; network coverage area = 64 mi^2, t_{max} =72 hours , Δt = 1 hour. The results are depicted in Figure 4. Note, the limits depicted in the plot represent the theoretical limits of reliability if the links were always existent (infinite transmission distance) and is therefore calculated from only the reliability of the nodes, via:

$$2TR_m(t) = P\{n_1(t) = 1 \cup n_n(t) = 1\} = r_i(t)^2 \tag{9}$$

$$kTR_m(t) = \sum_{i=k}^{n} \frac{n!}{i!(n-i)!} r_i(t)^i * (1 - r_i(t))^{n-i} \tag{10}$$

$$ATR_m(t) = r_i(t)^n \tag{11}$$

As with the previous method, this method may be utilized to understand the relationship between system (network) parameters and their impact on reliability. To demonstrate the additional utility of this model, the relationship between transmission distance and reliability is investigated. Table 3 shows the monotonically decreasing relationship of reliability and t_{ij}.

Figure 4. Mobility Model Results

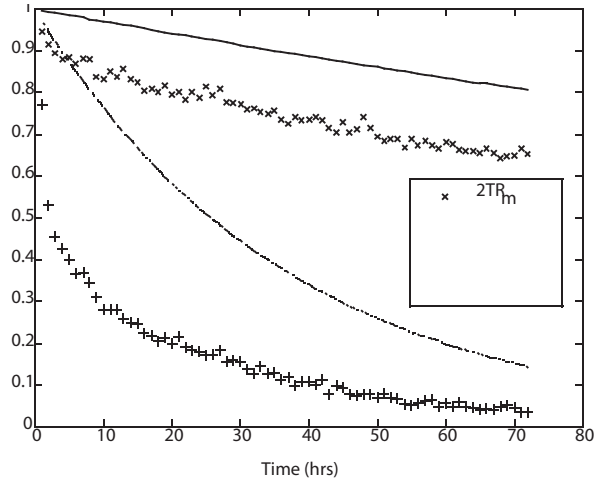


Table 3. Mobility Model Results

t_{ij} (miles)	$2\hat{T}R_m(72hrs)$
1	0.0444
3	0.6152
5	0.6846
7	0.7134
8	0.7652

Capacity Considerations

Removing the assumption of known probability of link existence provided the systems engineer more insight into reliability impacts of MAWN design parameters. This next method seeks to provide even more by further abstracting another assumption.

The assumption of constant transmission rate can be overly conservative or pessimistic depending upon how it is derived. It has been shown that transmission power, and by relation capacity, decreases over distance. Pahlavan and Krishnamurthy (2002, 224-229) described the phenomenon of wireless transmission and its degradation by path loss. Equation 12 illustrates the relationship between distance and received power; where d is the transmission distance; p_0 is the received power at a 1 meter distance; p_r is the received power at distance d; and α is the constant determined by the physical properties of the wireless medium; free space propagation yields $\alpha = 2$.

$$p_r = \frac{p_0}{d^{\alpha}} \tag{12}$$

The determination and optimization of capacity over a wireless link (channel) is a specialized field and a topic of much research and this paper does not attempt to advance this field but rather apply existing methods in combination with reliability analyses. Specifically, this paper combines the equation above with Shannon's Equation (Shannon & Weaver 1949). Shannon's Equation provides the theoretical maximum rate at which error-free digits can be transmitted and is therefore widely accepted as an appropriate model of the capacity of a wireless link, see Equation 13.

$$c = b * \log_2(1+s) \tag{13}$$

Within Shannon's Law, b is the bandwidth in Hz, and s is the signal to noise ratio (SNR). SNR at the receiver, with N_0 representing the noise at the receiver, is given by:

$$s = pr/N0 \tag{14}$$

It is also known that the demand upon a network's capacity is not constant.

Then, let the capacity demanded be $c_d(t)$ and the capacity available between a pair of nodes, i and j, be $c_{ij}(t)$. Therefore a link exists between nodes i and j if the available capacity c_{ij}, is greater or equal to the demand.

$$l_{ij}(t) = 1 \text{ if } c_{ij}(t) \geq c_d(t); \text{ else } l_{ij}(t) = 0 \tag{15}$$

To consider this, a relation between distance an capacity is developed using Shannon's Theory (Shannon & Weaver 1949).

$$\begin{aligned} & c = b * \log_2(1+s) \\ & s = \frac{p_r}{N_0} \\ & \Rightarrow c = b * \log_2(1+\frac{p_r}{N_0}) \\ & p_r = \frac{p_0}{d^2} \\ & \Rightarrow c_{ij}(d_{ij}(t)) = b * \log_2[1+\{(\frac{p_0}{d_{ij}(t)^2})/N_0\}] \end{aligned} \tag{16}$$

d_{ij} is the node separation distance
p_0 is the transmitted power
p_r is the received power
b is the bandwidth in Hz
s is the signal to noise ratio (SNR)

This method is then developed and implemented as the previous. Node mobility is again simulated via the RWMM. The distance between nodes is calculated in time and stored. Here, however, the distance is translated into capacity using the equation above. Next, the capacity demand is simulated from a probability distribution; in this case a standard normal distribution is used.

To do so, let the required capacity demand upon each link this metrics will be selected at random from a normal distribution defined by a mean (μ_d) and standard deviation (σ_d) in each time increment. The same network parameters as the previous example are used with the radio performance parameters (b, p_0, N_0) and mean capacity demands set to mimic the previous model characterized by transmission distance.

That is, b = 50 MHz; p_0 = 100 dB; and N_0 = 1 dB resulting in c_{ij}(3 mi) ≅ 320 bps.

The effect of capacity is noticed because the capacity is not fixed but based upon a distribution. Figure 5 shows that the impact of this volatility on capacity. In Figure 4 the reliability decreased smoothly as nodes failed with time by here there is variation due to the variance of the capacity demand.

This is even more pronounced when the standard deviation is increased from 25% to 100% of the mean, see Figure 6.

Figure 5. Capacity Model Results

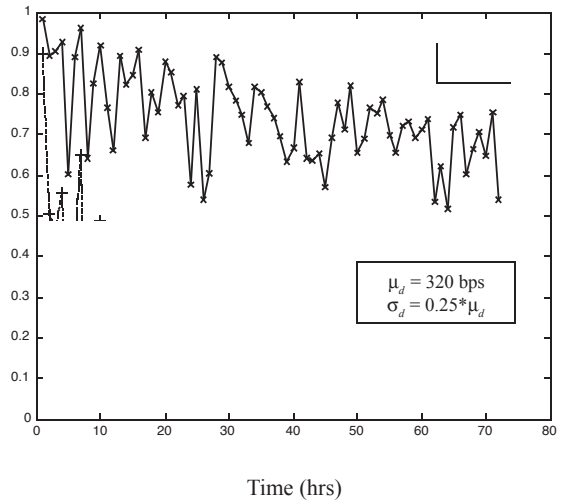
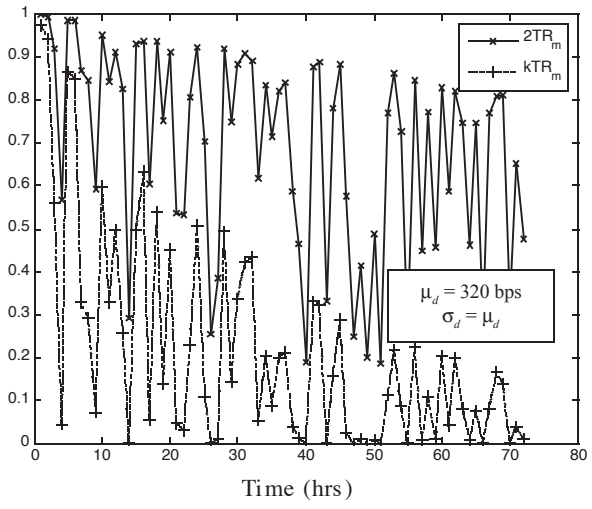


Figure 6. Capacity Model Results Future Work



This research is being done as part of doctorate program and is still on-going. Some near term expansion on this body of work that is planned follows.

A MAWN of practical scale often requires some segmentation, typically termed clustering, for successful implementation. The cluster-based network is best described as many smaller networks joined together by a back-bone network to form a single and cohesive network of networks. The network that joins the clusters together is usually known as a back-bone. Adaptation of these methods and development of more new techniques will strive to analyze the reliability of cluster based ad-hoc networks with a Monte Carlo based approach. Once completed, this will also provide insight into the affects to reliability due to design decisions on cluster size and gateway assignment.

This topic is also ripe for other research, an un-prioritized list of research areas that could provide meaningful contributions to the field are listed below.

Additional Research Areas
Design of Experiments to optimize network parameters
Application of Component Importance Measures for MAWN reliability
Application of machine learning techniques to make analysis of MAWN reliability more efficient
Application of these methods to other SoS metrics
Modification of methods to include additional mobility models
Inclusion of routing and MAC network layer considerations (i.e., inclusion of imperfection in route discovery and maintenance)

Conclusions

The contribution of this body of work is the capability these methods provide the system engineer. That is, the ability to define, understand and assess the reliability of this emerging network scheme based upon the attributes and parameters of the network. The methods developed allow for a complete systems view of MAWN reliability by providing quantitative methods to assess the impact of the system attributes that impact reliability. The system engineer is given the tools to analyze and optimize MAWN reliability and therefore the ability to influence and improve it.

In a systems context, these tools may be used to decompose high level network requirement to lower lever requirements for radio performance and reliability attributes.

The MAWN is still a break through technology; however, as it matures reliability will be expected and demanded by its users. Mostly, this work is motivated by the application of MAWN technology in the DoD tactical networks and the import of their reliable operation when employed for this use.

The results included herein and that will emerge from the continuation of the research agenda described will define the key contributors to MAWN reliability. The research performed to date has already identified the following key system parameters when considering reliability.

Key Reliability Drivers
Number of nodes in the network, n
Coverage area of network
Node reliability, r_i
Probability of link existence, λ
Capacity demands on the network, σ_d and μ_d
Performance of the nodes in terms of
Transmission range, t_{ij}
Performance of nodes in terms of transmitted power, p_0

In a systems engineering context, the methods provide the quantitative basis for requirement allocation and a trade space for the system (network) attributes analyzed. The same attributes identified above as reliability drivers may be concurrently "tuned" in order to develop an optimal or pseudo-optimal requirement set as a MAWN design solution.

References

Camp, T., J. Boleng, and V. Davies. 2002. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2*, no. 5: 483-502.

Chen, Zinyu and Michael R. Lyu. 2005. Reliability analysis for various communication schemes in wireless CORBA. *IEEE Transactions on Reliability 54*, no. 2: 232-242

Cook, Jason L. and Jose E. Ramirez-Marquez. 2006. Two-terminal reliability analyses for a mobile ad-hoc wireless network. Accepted for publication by Reliability Engineering and System Safety.

Ebeling, Charles E. 1997. An introduction to reliability and maintainability engineering. Waveland Press.

Freebersyser, James A. and Barry Leiner. 2001. A DoD perspective on mobile ad hoc networks. In Ad hoc networking. chap. 2: 29-51.

Pahlavan, Kaveh and Prashant Krishnamurthy. 2002. *Principles of wireless networks.* Prentice Hall PTR.

Ramirez-Marquez, J.E. and D. Coit. 2004. A heuristic for solving the redundancy allocation problem for multistate series-parallel systems. *Reliability Engineering & System Safety 83*, no. 3: 341-349.

Rocco, Claudio M. and Marco Muselli. 2004. Empirical models based on machine learning techniques for determining approximate reliability expressions. *Reliability Engineering and System Safety 83*, no. 3: 301-309.

Shannon, C.E. and W. Weaver. 1949. The mathematical theory of communication. Urbana, IL: University of Illinois Press.

Biographies

Jason L. Cook is a Reliability Engineer for the Quality Engineering and System Assurance Directorate of the Armament Research and Development Engineering Center at Picatinny Arsenal, NJ. In this capacity, Jason is responsible for the reliability programs on computing, communications, and sensor systems developed for the US Army and Joint DoD services under the Future Combat Systems program. In addition, Jason is a member of the American Society of Quality and has met their requirements for certification as a Certified Reliability Engineer (CRE). Jason has recently received a Ph.D. in Systems Engineering from the Systems Engineering and Engineering Management Department of Stevens Institute of Technology. His research interests include network reliability with a specific focus on applications to DoD networks and networked systems. He currently has 4 published works in this area and two pending.

Jose E. Ramirez-Marquez is an Assistant Professor at Stevens Institute of Technology in the Department of Systems Engineering and Engineering Management. In this capacity, he is serving as the advisor for Jason's research. Jose's research interests include system reliability and quality assurance, uncertainty modeling, advanced heuristics for system reliability analysis, applied probability and statistical models and, applied operations research. He has authored more than 12 articles in leading technical journals on these topics. He obtained his Ph.D. at Rutgers University in Industrial and Systems Engineering. He received his B.S. degree in Actuarial Science from the UNAM in Mexico City in 1998 and M.S. degrees in Industrial Engineering and Statistics from Rutgers University. He is a member of IIE, IFORS and INFORMS.

Alaska Airlines Flight 261: Understanding the Systemic Contributors to Organizational Accidents

Christian G.W. Schnedler, Daniel Murphy, Steven J. Stumpp, Frantz St. Phar
Stevens Institute of Technology

Introduction

On January 31, 2000, at approximately 16:21 Pacific Standard Time, Alaska Airlines Flight 261 crashed into the Pacific Ocean off the California coast just west of Los Angeles. The crash killed all 88 passengers and crew members onboard. After an extensive investigation by the National Transportation Safety Board (NTSB), the cause of the accident was attributed to a failed jackscrew assembly controlling the horizontal stabilizer in the tail section of the airplane. This caused the plane to pitch nose-down, rendering it completely uncontrollable once the jackscrew failed.

Factors leading to the crash of Alaska Airlines Flight 261 uncovered in the NTSB report included Federal Aviation Administration (FAA)-approved lengthened inspection intervals; the use of unapproved tools and methods of measurement for checking the jackscrew assembly and assessing it for wear (endplay check); falsifying maintenance reports to show work had been completed when none took place; receiving approval for maintenance manual and procedural changes without consent from the FAA, director of base maintenance, or the director of maintenance planning and production control; and various interpretations by mechanics at different repair/inspection facilities without regards to proper inspection procedures of the jackscrew assembly.

We will analyze this crash, utilizing principles and organizational theories described by Reason (1997) which focus not on the technical failure of the mechanical components, but on the roles played by the human influence from upper management of Alaska Airlines and the FAA down to the culture of the maintenance crew involved. This analysis paints a clear picture of how minimal importance was given to safety in this organization and how unmonitored practices eventually breached the well-intentioned, but unjustifiably neglected, systemic defenses in place.

We summarize that the root cause of Flight 261's tragic end was not the failure of the jackscrew assembly, but rather the cumulative effect of both economic and organizational pressures acting on all levels of Alaska Airline's organizational hierarchy. We further propose that the true value of the lessons learned from Flight 261 lies in the importance of taking a comprehensive, systems perspective of organizational risks. Finally, we cite the Tripod-Delta Model as an example of a systems-based risk mitigation tool, though we also note that the need remains for more advanced tools capable of systematically mitigating core organizational risks identified.

Alaska Airlines Flight 261 departed from Puerto Vallarta, Mexico at 1:37pm on January 31, 2000. Two hours and forty-four minutes later it would crash into the ocean off the coast of California just west of Los Angeles. Following the crash, the National Transportation Safety Board (NTSB) traversed through standard protocol: examining the wreckage, interviewing maintenance crewmembers, pilots, and executives from Alaska Airlines, the FAA, and even NASA; and determined the cause of the accident to be "a loss of airplane pitch control resulting from the in-flight failure of the horizontal stabilizer trim system jackscrew assembly's acme nut threads."

While this approach provided a tangible error able to be remedied with additional, stringent regulations and standards, it focused attention solely on maintenance practices and standards; effectively placing the majority of the blame on lubrication intervals rather than considering the underlying, systemic contributors to the tragedy.

Case Overview

On January 31, 2000, Alaska Airlines' Flight 261, an international passenger flight traveling from Diaz Ordaz International Airport (PVR) in Puerto Vallarta, Mexico to Seattle-Tacoma International Airport (SEA) in Seattle, Washington was to transport a total of eighty-eight passengers and crew to their destinations on a McDonnell Douglas MD-83 aircraft. The flight departed PVR at 13:37 Pacific Standard Time (PST) en route to its scheduled stopover at San Francisco International. At approximately 15:49 PST, the captain of Flight 261 contacted Air Traffic Control (ATC) at SEA requesting permission to divert the flight to Los Angeles International Airport (LAX) due to a jammed horizontal stabilizer. At 15:57 PST, the captain deemed landing at LAX absolutely critical due to weather

and flight conditions. The captain then relayed his decision to ATC at SEA and requested an open channel to LAX ATC.

En route to LAX, at 16:07 PST, the flight crew began discussions with a LAX Alaska Airlines maintenance worker. A series of maintenance checks on the horizontal stabilizer and primary trim motor electric circuit breakers were performed during the five-minute conversation between the LAX Alaska Airlines maintenance worker and the flight crew. During these checks, unfamiliar noises were heard emanating from the aircraft. Shortly thereafter, the plane plummeted from an altitude of approximately thirty-one thousand feet to twenty-four thousand. The captain notified the maintenance worker of the aircraft's rapid altitude decent immediately after stabilizing the plane. Unable to explain what was causing the unusual noises, the maintenance worker suggested that the flight crew perform the same troubleshooting checklist for the horizontal stabilizer and primary trim motor circuit breaker at their own discretion.

The flight crew did not adhere to this request due to flight conditions encountered from the previous checks. Shortly thereafter, at 16:15, one of the flight crew members contacted a Los Angeles Air Route Traffic Control Center (ARTCC) controller requesting a descent altitude for preparatory maneuvers for landing. The ARTCC controller granted them an altitude of seventeen thousand feet and directed the flight crew to another ARTCC controller.

After receiving a new block altitude, heading, and frequency at 16:17, the last contact was made between an outside agent and the flight crew. In an attempt to slow the airplane and decrease altitude, the plane pitched nose-down and rolled over 180°. The captain and first officer tried in vain to right the plane, but their efforts failed. Flight 261 crashed January 31, 2000 at approximately 16:21 in the Pacific Ocean, 2.7 miles north of the Anacapa Islands, California. An investigation by the National Transportation Safety Board (NTSB) would later find the accident resulted from a failed jackscrew assembly controlling the horizontal stabilizer (see Figure 1).

During the period before the crash, the official industry documentation maintenance procedure was the Maintenance Steering Group 2 and 3 (MSG-2, MSG-3). The MSG-3 document contained decision logic and procedures for use in maintenance and inspection programs which coincided with the Federal Aviation Administration (FAA) requirements. The MSG-3 document was created to reduce the complexity of understanding the MSG-2 document and provide clear and concise guidelines on how to interpret the maintenance process outline. The decision logic behind MSG-3 is the cascading failure approach; better known as a "consequence of failure approach."

In 1985, Alaska Airlines released its own maintenance procedures in compliance with the guidelines in both the MSG-2 and MSG-3, but with stricter requirements. This document was known as the Alaska Airlines' Continuous Airworthy Maintenance Program and was approved by the FAA. This document listed time intervals stating when routine scheduled inspection and maintenance intervals on the planes shall be done. As years passed, Alaska Airlines adjusted this document without notifying the FAA in an effort to improve performance; focusing on meeting self-appointed criteria rather than the industry standards (see Table 1 for the chronological adjustments made to of Alaska Airlines' lubrication intervals).

Figure 1. Jackscrew from Flight 261 Horizontal Stabilizer (NTSB)

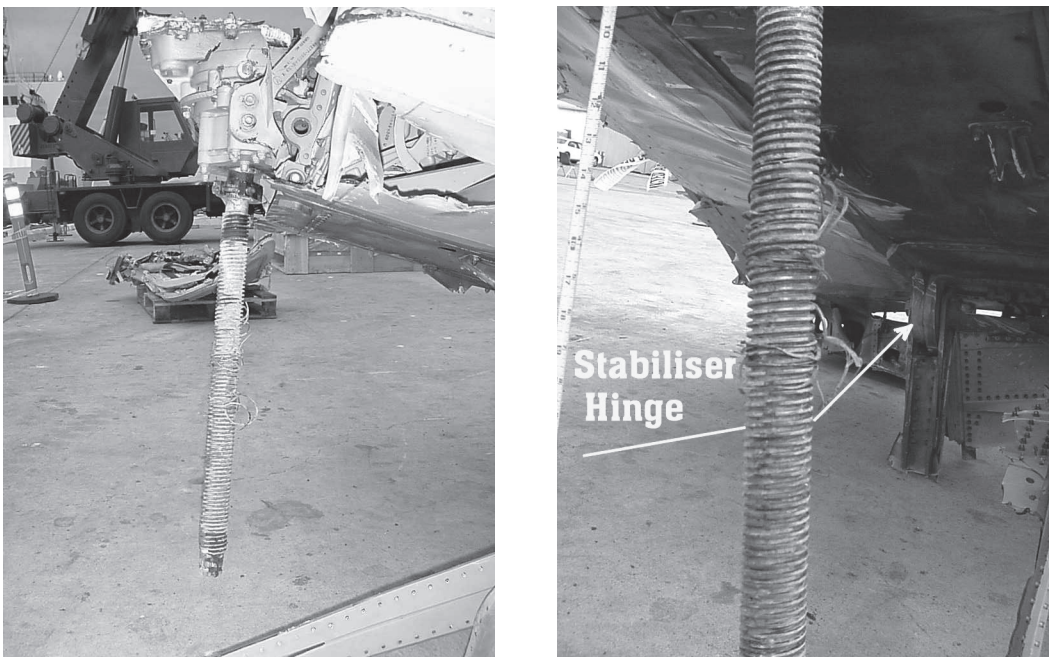


Table 1. Comparison of Jackscrew Assembly Lubrication Intervals (NTSB 2002)

MSG-2 MRB	MSG-2 OAMP	MSG-3 MRB	MSG-3 OAMP
		C check	C check
Not included in logic diagram	600 to 900 flight hours	(3,600 flight hours or 15 months, whichever comes first)	(3,600 flight hours)

Alaska Airlines 1985	*Alaska Airlines 1987*	*Alaska Airlines 1988*	*Alaska Airlines 1991*	*Alaska Airlines 1994*	*Alaska Airlines 1996 to April 2000*	*Alaska Airlines April 2000 to Present*[a]
Every other B check	B check	Every eighth A check	Every eighth A check	Every eighth A check	Time-controlled task card - 8 months maximum	650 flight hours
(700 flight hours)	(500 flight hours)	(1,000 flight hours)	(1,200 flight hours)	(1,600 flight hours)	(About 2,550 flight hours)	

a. All carriers currently meet this requirement

The National Transportation Safety Board's Investigation Report uncovered a vast array of maintenance red flags prior to the crash of Flight 261. It was this lack of proper maintenance, extended intervals between inspections, and possible missed lubrication intervals due to falsified work reports that led the NTSB to conclude the accident was the result of a waterfall of maintenance errors.

Case Study Approach

Within any organizational system, numerous underlying, potentially dangerous conditions exist capable of creeping into an organization's culture if left unmonitored. We believe it is the responsibility of the organization to ensure measures are in place to either intercept such risks prior to their becoming catastrophic accidents or, at least, minimize the damage created by these conditions' breach of the system's defenses. More specifically, we will explore and explain organizational issues related to:

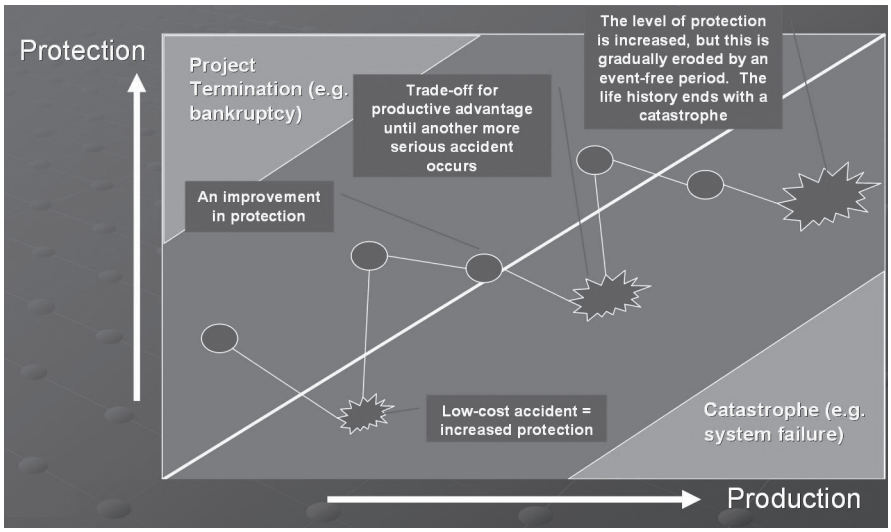
- The factors, outside of the technical failures, which combined to cause the crash of Flight 261.
- The identification of the key stakeholders and how they failed to recognize the system's warning indicators.
- How other organizations can utilize the lessons learned from Flight 261 to help prevent accidents of this magnitude from happening to them.

Far too often, analysis of major accidents concentrates on the final operators, mechanics, or points of failure in a system; it is always easier to place blame on an individual or group than it is to find fault with an organization as a whole. However, as explained by organizational theorists such as Reason (1997) (also see Perrow 1999), it is often these underlying, organizational systemic conditions which are most responsible for creating an environment in which disaster is all but inevitable. Applying this logic to Alaska Airlines Flight 261, we believe it is precisely such pervading, system-wide conditions which ultimately led to the flight's horrific end.

In general, the effect of the dynamic relationship between external and internal pressures leading to organizational accidents can be explained as a continuum between performance and safety organizations operate in (see Figure 2). During its lifetime, an organization faces constant pressure by both external and internal stakeholders to increase performance and improve on its metrics of success (in business organizations, this is typically akin to "the bottom line"). As explained by Reason, the result of this drive for performance is a tradeoff in safety.

At the top level of organizations, this tradeoff is often found in a marked decrease in risk aversion (for example, pursuing riskier opportunities with hopes of greater rewards). For middle and lower management, the increased emphasis on performance at the expense of safety is most often found in the reallocation of resources away from maintenance, security, and other safety measures and towards improving the effectiveness and overall output

Figure 2. The Production-Protection Space (Reason 1997)



of operations. Especially in business, this phenomenon is built into the nature of the system. Maintenance and safety offer no immediate, tangible return on investment; risk avoidance is extremely hard to quantify, and therefore increasingly hard to justify in times of strict budget constraints.

Initially, organizations are established with various layers of explicit and inherent defenses to help prevent the natural dangers of a system from creating accidents. However, over time, the aggregate effect of this tradeoff of safety for performance results in the formation of holes in the system's defenses. If these holes are left unchecked, it is only a matter of time before they will align themselves and allow the latent conditions to breach the system's defenses.

For a simplified example of this phenomenon, consider the following scenario of a typical small business: Initially, defenses are erected via training of operators, maintenance procedures, and an external regulatory agency. Within the company itself, there is pressure on the operators to perform at ever-higher levels. Further, the pressures to reduce costs eventually lead middle managers to shift resources away from maintenance (through decreased budgets and/or a reduction in personnel). Meanwhile, the external regulatory agency faces similar pressures (usually during a protracted time without major accidents); leading the agency to become more lax on its checks and potentially experience budget cuts and reductions in personnel as well.

In such a scenario, it easy to see how the danger inherent in day-to-day business operations will escalate as the pressure for increased performance continues to rise. Further, the defensive barriers provided by the maintenance staff and regulatory body will progressively succumb to other organizational pressures and eventually erode away. In time, this decrease in defenses and increase in danger will lead to a complete breach of defensive layers and result in either an organizational accident or near-accident (described by Reason as occurring when one of the final layers of defense prevents an accident through extraordinary measures). See Figure 3 for a visual of this phenomenon.

Once an accident or near-accident occurs, new defenses will be erected, policies put in place, and personnel hired. However, as described by the performance-safety continuum above, this emphasis on safety will, in time, give way to performance and the cycle will continue. In his book, Reason offers numerous historical examples of such phenomena occurring in industries as diverse as nuclear power and space exploration. Alaska Airlines Flight 261 also stands out in its similarities, both in terms of underlying systemic issues and ultimate outcome, to these catastrophic accidents resulting from failures in the organizational system.

Case Analysis

While placing blame on tangible, technical malfunctions is much easier to understand, it rarely addresses the root cause of the breakdown in a system's defenses. Reason discusses Pareto's Rule (the 80:20 rule); concluding that 80% of accidents can be traced back to human failures, while only 20% are the result of technical malfunctions. As the NTSB report reveals, sufficient defenses were originally put in place to prevent the series of events leading to the crash of Flight 261 from ever occurring. However, due to a series of human errors and "culture creep" spanning all echelons of the organization, by the time of the accident these defenses had been left unmonitored, circumvented, and stretched too thin to function effectively. See

Figure 3. The "Swiss Cheese" Model of Defense Layers Breached (Reason 1997)

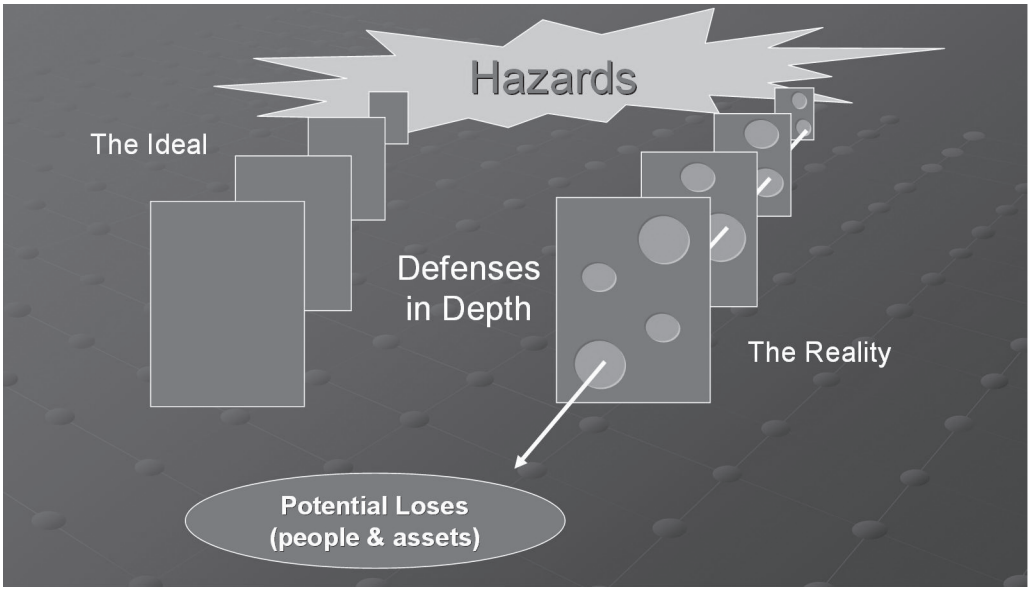
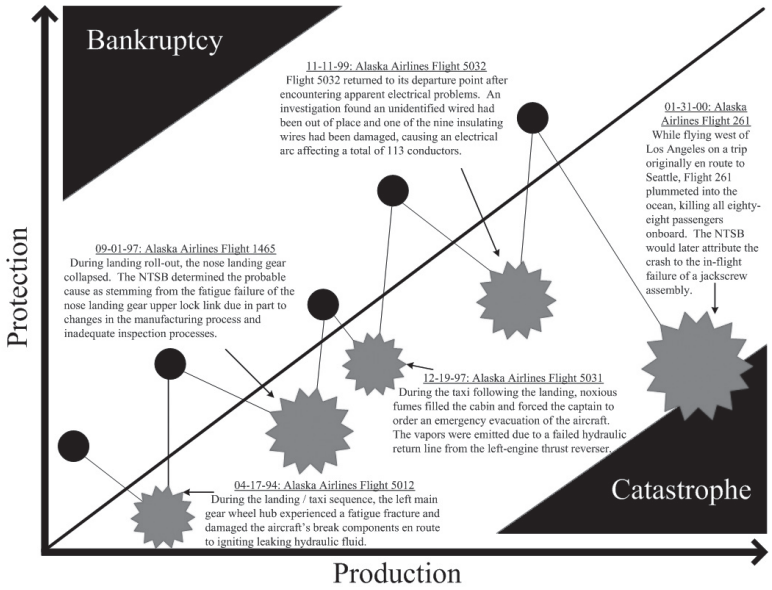


Figure 4 for an adaptation of Reason's Production-Protection Space representing the progression of Alaska Airlines along the production-protection continuum.

In order to determine the root cause of the Alaska Airlines accident, one must first consider the environment in which the airline operated. Acting as its own system of checks and balances, the Federal Aviation Administration serves as a watchdog over the airline industry. The airlines themselves are under constant pressure by the forces of a capitalistic market, with effective management of time and money providing the fundamental basis of their business success. These pressures were compounded by the aggressive expansion Alaska Airlines was undergoing just prior to the Flight 261 crash. To keep pace with the market, Alaska needed as many airplanes operational at a given time as possible. As explained in the Approach section above, this constant pressure for increased performance, coupled with the lack of tangible results safety measures are able to provide, created a focal point of pressure at Aviation Management Systems (AMS), the organization responsible for providing maintenance work to Alaska and other airlines (Miletich 2001).

Positioned at the highest level of the aviation industry, the FAA is responsible for overseeing every public airport in the United States, and therefore every airline and airplane manufacturer (including parts manufacturers). As a result, the organization's means of providing oversight is comprised primarily of stringent manuals for operations,

Figure 4. Alaska Airline's Production-Protection Space (NTSB Accident Reports)



maintenance, and other protocols sent to each downstream entity. However, the authority of the FAA is very limited; they have the responsibility of providing oversight without the power to enforce their decisions. This lack of explicit authority thus places the onus on the airlines and airplane manufactures themselves to incorporate the FAA policies into their individual General Maintenance Manuals (GMM). This passive regulatory stance is evident throughout all layers of the governmental organization, and is even found within its mission statement:

> "We provide technical and *advisory guidance* on airport planning and development; we inspect airports to help assure the safety of airport operations; we are responsible for environmental assessments of proposed construction and approval of noise compatibility programs; and we administer the Airport Improvement Program (AIP) and the Passenger Facility Charge (PFC) program. We also *monitor* airports to assure protection of the federal investment. We work extensively with airport owners, airport users, the aviation industry, and state and local governments to provide a safe and efficient system of airports for all who fly in the United States of America."

In the case of Flight 261, the systemic problems resulting from the lax oversight of the FAA is evident in numerous conversations with individuals involved with Alaska Airlines and Aviation Management Systems following the tragic crash. For instance, an FAA audit of Alaska Airlines after the highly-publicized accident found serious deficiencies in Alaska's maintenance program that had existed for months and even years before the crash, but went undetected by the FAA's regional headquarters in Renton, Washington. (Miletich 2001) This likely stemmed from the fact that FAA technicians neither have authority under FAA regulations to sign off on work completed nor work side-by-side with AMS mechanics and inspectors (Miletich 2001)—even though Alaska Airlines had explicitly requested an increase in FAA presence to meet the increased number of inspections required by their growing operations. (NTSB 2002)

As research into Alaska Airlines operations revealed, problems with the FAA's oversight approach impacted the airline industry far beyond the failure to adequately inspect maintenance operations; directly influencing the subsequent culture of aviation industry manufacturers, airlines, and maintenance workers. For example, the inspection of Flight 261 uncovered that one reason the damage to the jackscrew assembly was not recognized was that the tools used to analyze the assembly were created in-house by the maintenance staff themselves, and were subsequently not as accurate as manufacturer-made models. When questioned about this, the FAA responded that "the determination of equivalency for such equipment is the primary responsibility of the repair station or the air carrier, not the FAA" (NTSB 2002).

There is also evidence that the lack of meaningful oversight by the FAA led to complacency within the maintenance culture when dealing with suggested rules and regulations. Asked by an FAA attorney how supervisors could sign off for work done by mechanics, even when they hadn't performed the work themselves and when Alaska's maintenance manual states in capital letters that problems "MUST" be corrected and signed by the person doing the work, the maintenance manager questioned replied: "It doesn't say you can't" (Miletich 2001).

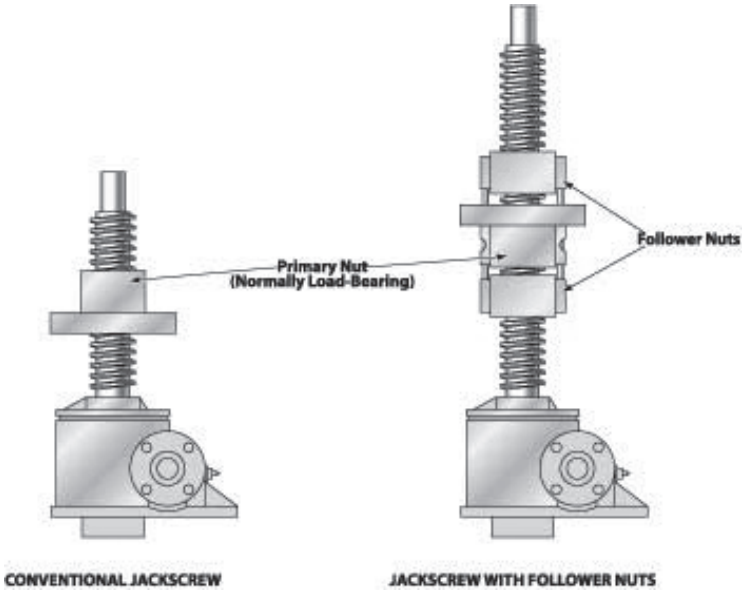
Even though such failures in the FAA's oversight clearly contributed to the fateful accident of Flight 261, to attribute all of the blame to the organization is neither fair nor representative of the multitude of forces involved. For instance, latent conditions contributing to the crash can be traced to the original manufacturer of the jackscrew assembly: Peacock Engineering (it has since been acquired by Trig Aerospace). (KSC Support) Although the company itself was not directly involved with the accident—the jackscrew assembly had been installed eight years prior to the plane's crash—the same underlying economic forces which impacted the FAA affected this manufacturer of jackscrew assemblies; ultimately leading it to continue producing and promoting a product without built-in redundant defenses capable of guarding against lapses in maintenance.

In 1998, nearly two years prior to the tragic crash of Flight 261, engineers at NASA's Kennedy Space Center (KSC) were made aware of the consequences of possible jackscrew failures during an incident involving the gaseous oxygen (GOX) vent arm. Even though it had already been prepared for the next launch, technicians at KSC decided to perform an additional test to verity proper arm alignment with the external tank (ET). During the test, the jackscrew nut threads sheared and the GOX hood fell from its position. If the failure had occurred on the next planned cycle, severe damage would have been sustained by the shuttle vehicle (KSC Support).

Alarmed, KSC formed a team to design an improved jackscrew assembly able to be more easily monitored by maintenance crew members and retrofitted with a fail-safe feature in case of damage to the primary jackscrew. This crew would find the solution to these objectives in a design based on redundant follower nut(s) as shown in Figure 5.

After devising the new assembly, this same task force was charged with determining whether a commercial market existed for the improved design. They quickly found that the pressures for economic performance constantly at work in commercial industries led only one of the manufacturers contacted to indicate a desire to consider licensing the improved design. As the group's findings describe: "Most modern commercial use of jackscrews occurs in applications where failure does not

Figure 5. Original and Redesigned Jackscrew Assembly (Fraley et al.)



physically endanger individuals. The majority of companies producing jackscrews and ballscrews were not interested in safety technologies for jackscrews…. No market drivers are apparent…" (KSC Assessment 2001).

Much like the FAA, the manufacturers of jackscrew assemblies placed the onus of maintenance and safety on the individual airlines rather than themselves. As described by an individual associated with the design of the jackscrew: "The major jackscrew manufacturers… (did not solve) the problem because they did not recognize it as their problem… Sentiments (were heard) that (the manufacturers) produce the jackscrew and the user must maintain it, and if the recommended maintenance procedures are followed then failure is unlikely" (KSC Assessment 2001). However, the NTSB investigation following the crash of Flight 261would result in the grounding of twenty-seven of Alaska Airline's jets due to potential problems with the jackscrew mechanism (KSC Assessment 2001). Clearly, passing responsibility to the airlines themselves was not an adequate safety solution.

Within Alaska Airlines itself, the systemic issues leading to the crash of Flight 261 seem to be the result of long periods of relatively safe operations leading the organization to adopt a culture centered on performance at the expense of safety. According to a panel of safety experts hired by Alaska Airlines to scrutinize its operations, there were "no glaring safety deficiencies. (Alaska Airlines) had all the programs and all the procedures in place, but the safety elements of the airline were too diffused" (Ayer 2000). However, we believe this observation is a natural outcome of the progression of Alaska Airlines along the performance-safety continuum. As noted by the analysts, all of the necessary defensive barriers were firmly established in Alaska's system. However, over time, the focus on safety gradually gave way to the need for performance, leading the "safety elements" to become diffused and ineffective. This concept is accurately described as "culture creep" by Enders Associates International as follows:

> "'Culture creep' can evolve into a rationale for operating beyond regulatory intent with, for example, deferred maintenance, excusing 'minor' procedural non-compliance on the flight deck and in ground operations and other procedures, etc. Conformity with a company's own stated policies and procedures can also be insidiously eroded if 'culture creep' is permitted to persist."

Much as with both the FAA and jackscrew assembly manufacturers, the economic pressures inherent in the airline market combined with a gradual shift in culture; ultimately leading to a sacrifice of safety for performance within Alaskan Airlines. This progression along the performance-safety continuum and onset of culture creep was ultimately focused within Alaska Airline's maintenance division. During its analysis of the various practices employed by Alaska's maintenance staff, the NTSB found startling discrepancies between the procedures outlined by FAA regulations, Boeing manufacturer data, and even Alaska Airlines' own General Maintenance Manual (GMM) and what was perceived as acceptable practices by the maintenance crew.

Relative to the jackscrew assembly, the NTSB found that maintenance facilities were using shop-made tools to perform the invasive end play check inspections to determine thread wear relative to the nut's design wear limit. When questioned on the tools and the procedure to determine end play, Alaska maintenance crew told the board that they would continually measure and re-

measure the jackscrew end play with the wrong tool until the "right" answer (within tolerance) was produced. At 40 thousandths of an inch slack, the assembly was within tolerance. At 41 thousands of an inch, the end play was deemed excessive and the jackscrew and acme nut had to be replaced with a matched pair (Air Safety Weekly 2002).

As described above, the Alaska maintenance crew routinely made their own tools to perform the end play checks rather than purchasing the more accurate, but also more expensive, Boeing-manufactured tools. When questioned about the use of this shop-made instrument, an Alaska Airlines' manager of tool control told investigators that "what the maintenance staff members were making 'wasn't even close' to Boeing's engineering drawing requirements," and that "we were directed to build the tools, and we did exactly what we were told" (NTSB 2002).

To further spotlight the extent to which organization-wide, systemic issues affected the culture of Alaska Airline's maintenance unit, consider the following examples of common maintenance practices uncovered by the NTSB investigation:

- Substituting Aerosol 33 for Mobilgrease 28 before FAA approval and having it receive Alaska Airlines Reliability Analysis Program Control Board approval without the required signatures of the director of base maintenance or the director of maintenance planning and production control.
- Mixing Aerosol 33 with Mobilgrease 28 with no lab data saying it was safe to do so (non-corrosive to the nut and/or jackscrew metals).
- Signing off on work that is not yet complete. (A senior Alaska Airlines mechanic admitted in court that supervisors regularly sign off on maintenance work that has not been completed.) (Channel 600)
- Performing maintenance in far less time than specified in Boeing's maintenance procedures (4.5 hrs vs. "a couple hours" at the Oakland maintenance facility vs. "approximately an hour" at the San Francisco maintenance facility).
- Maintenance crew admitting they did not know the correct procedure to maintain, measure, and lubricate the jackscrew assembly.
- Successfully petitioning the FAA to extend total maintenance C-Check intervals by 200% between 1985 and 1996 (see **Case Overview**). This inadvertently extended specific task end play check intervals to beyond acceptable levels (every 30 months, or ~9,550 hrs).
    - From the last end play check inspection in September 1997 to the crash, the wear rate of the nut threads was roughly 10 times what was expected with regular maintenance and use. Upon recovery of the wreckage, the acme nut threads showed wear of 90%. At the nut's wear limit (when the nut should be replaced) it should exhibit wear of 22%.

Conclusion

In its report, the National Transportation Safety Board (NTSB) identified the crash of Alaska Airlines Flight 261 as resulting from a failed jackscrew assembly. The blatant maintenance lapses within the airline were further noted as the primary contributor to the flight's tragic end. However, although the NTSB's report accurately described *what* happened, it could not explain *why* it happened. As the preceding analysis has described, the tragic end of Flight 261 was not the result of a sole failure by the maintenance staff to correctly diagnose the jackscrew's condition. Rather, the accident was the result of the combined pressures of economic forces and a period of incident-free flights gradually eroding away the systemic defenses built into Alaska Airline's operational system and facilitating breaching of these defenses by pervasive, latent conditions.

Having failed to address these systemic variables, the recommendations in the NTSB report have had little effect on any parties belonging to the airline organization. Comments from the NTSB members indicate that even after the accident, nothing has changed from the organization's attempt to improve Alaska Airlines' safety practices. The NTSB suggested that the FAA inspect the airline to evaluate whether "adequate measures have been fully implemented to sure the deficiencies identified in the FAA's April 2000 special inspection report" (NTSB 2002). This did not take place; the FAA cited an inability to divert already-stretched resources from other important tasks.

As further proof of the lack of adequate safety measures taken since Flight 261, as Alaska Airlines Flight 506 climbed above 10,00ft on March 25, 2000—two months after the crash of Flight 261—the plane failed to pressurize and the oxygen masks deployed. As the passengers began to use the masks, the pilots found they quickly depleted the emergency oxygen on board. The flight continued to its destination without injury, but the legal ramifications resulted in the flight's pilot losing his license for continuing to fly with no emergency oxygen left. It was later discovered that the pilots failed to notice that a "bleed air" switch was mistakenly left in the OFF position after it had been checked by maintenance and recorded as being placed back in the ON position. In this case, the ramifications were concentrated on the pilot and no actions were taken against the maintenance crew.

Even if actions had been taken against the maintenance crew—in the case of prior accidents, including Flight 261, such actions were—there would have been little lasting impact. Similarly, the answer to the issues plaguing jackscrew assemblies can not be resolved with the addition of an output device able to detect wear (as suggested by the Kennedy Space Center team which developed the redundant follower-nut design). (KSC) While such remedies are suitable for accidents arising solely from gross negligence

or incompetence on the part of a few, key operators at the final interface of a system, they fail to address such systemic issues as those affecting Alaska Airlines. Rather, the correct measures for overcoming the systemic actors contributing to the tragic end to Flight 261 must come from the organization's safety culture as a whole.

As expressed in the Approach section above, there are many forces inherent in capitalist business environments which resist *organizational* safety systems. When budgetary cuts are required, the first program affected is often the safety program. It is also the last program to receive additional funding when a company is experiencing profitable growth. As a result, cutting safety corners in an effort to increase the bottom line is a common attribute of unregulated industries. For Alaska Airlines, this unregulated shift is clearly evident in Judge Patrick Geraghty's frustration during a trial on the falsification of the airline's maintenance records in 1998: "(The flaws in the defense system exist) because the entire maintenance system is an honor system," the judge said. "So if the records aren't accurate the whole system collapses. And that certainly affects the flying public and air safety" (Miletich 2001).

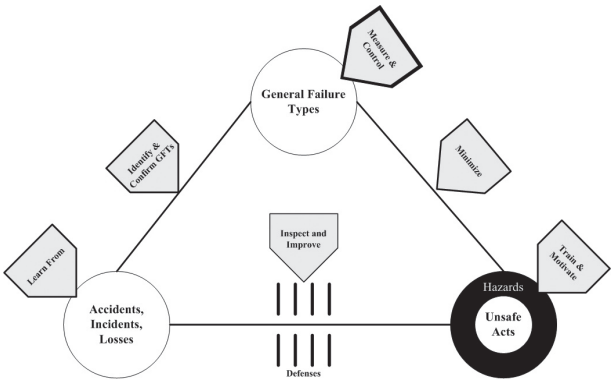
To overcome the dangers inherent in the airline industry, the NTSB recommendations should have focused on implementing organization-wide safety management systems to ensure that as holes develop in the various defensive layers, they are recognized, confronted, and repaired. The fallout of Flight 261 itself included recommendations similar to this approach, such as the view from John Enders and William Hendricks in their "Safety Assessment" report: "Essential to effective risk management is a risk assessment process by which risk can be identified, measured, evaluated and controlled. In other words, safety should be viewed as *a core production value* of the organization and, as such, a value that will accrue to the benefit of the airline, its employees and to its customer base. What better reputation could be forged than a solid, credible acceptance by the customers of the airline as a safety leader in commercial aviation?" (Enders and Hendricks 2005)

An example of such a safety management system currently in place to combat the systemic eroding of defenses can be found in Shell International Exploration and Production BV's Tripod-Delta project. Beginning in 1998, the Tripod project was developed around three core elements (Reason 1997):

- A coherent safety philosophy that leads to the setting of attainable safety goals.
- An integrated way of thinking about the processes that disrupt safe operations.
- A set of instruments for measuring these disruptive processes—termed General Failure Types (GFTs)—that does not depend upon incident or accident statistics (that is, outcome measures).

Before its introduction, Shell's principle safety metric was the number of lost-time injuries per million man-hours (LTIF). However, this system was only effective at diagnosing accidents ad hoc. Tripod-Delta, on the other hand, focuses on General Failure Types: the situational and organizational factors which, without intervention, would inevitably lead to lost-time injuries. See Figure 6 for a general overview of how the Tripod-Delta program operates.

Figure 6. Tripod Delta—Examining Types of Failures & Learning to Prevent Them (Reason 1997)



Similar practices can also be found in governmental agencies such as the Occupational Safety and Health Administration (OSHA). The very mission of OSHA's is to assure the safety and health of America's workers by setting and enforcing standards; providing training, outreach, and education; establishing partnerships; and encouraging continual improvement in workplace safety and health. Much like Tripod-Delta, this approach focuses not on preventing past injuries from being repeated, but preventing *future* types of accidents from ever occurring.

Herein lies the true lesson learned from Flight 261. So long as risks and accidents are viewed as singular events in need of correction, the underlying, pervasive conditions which facilitated their breaching of organizational defenses will remain unchanged. It is only after a more comprehensive, systems perspective is adopted can the true stimuli be uncovered. Utilizing such tools as the Tripod-Delta Model, these risks can then be overcome with standard mitigation techniques and other, yet undiscovered, tools capable of systematically mitigating the core organizational risks identified.

References

Ayer, Bill. 2000, July 6. *Alaska's World*.

Channel 6000. 2001, May 2. Airline Mechanic Admits Falsifying Work Records.

Conrard, Don. 2001, 31 January. Metal of honor: Flight

261 Pilots Earn Rare Commendation From ALPA. *People*.
Enders and Hendricks. 2005, Nov. 9. Sampling of Observations and Recommendations from the Enders and Hendricks. Safety Assessment. Retrieved May 2006 from: *http://www.iasa.com.au/folders/Safety_Issues/others/AlaskaAssessed.html*.
Federal Aviation Administration. Mission Statement: Airports: Northwest Mountain Region." Retrieved May, 2006 from: *http://www.faa.gov/airports_airtraffic/airports/regional_guidance/northwest_mountain/about_airports/mission_statement/our_mission/*.
FindArticles.com. Pilots Accused of Endangering Passengers. Retrieved May 2006 from: *http://www.findarticles.com/p/articles/mi_m0UBT/is_37_14/ai_65176525/pg_2*.
Fraley, J., I. Valez, and C. Stevenson. Resolving A Lack of Redundancy. *NASA Tech Briefs: KSC-12187/291/92*.
Hager, Robert. MSNBC TV News. FAA Checks Airlines For Maintenance. 9 Nov. 2005.
Innovative Technology Institute (ITI). KSC Technology Assessment. 22 May 2001.
Miletich, Steve. Airline Tool, Similar To Piece Tied To Alaska Crash, Mysteriously Appears. 27 April 2001.
Miletich, Steve. 2001, May 1. FAA Proposes Alaska Air Fine; Judge Revokes Mechanic's License.
Miletich, Steve. 2001, May 3. Flight 261 case Far From Over.
Miletich, Steve. 2001. Inspection Problems Cost FAA Official His Post.
Miletich, Steve. 2001, April 25. Plane-Repair Problems Continue At Alaska Air.
Miletich, Steve. 2001, April 29. Questions Over Alaska's Repairs: Conflict Noted Between Airline's Policy, Practice.
National Aeronautics and Space Administration. Fail-Safe, Continue-to-Operate Concept for Jackscrews. *NASA Tech. Briefs: KSC-12187/291/92*.
National Transportation Safety Board. American Airlines, Inc. DC-10-10, N110AA Chicago-O'Hare International Airport Chicago, Illinois May 25, 1979. Retrieved May 2006 from: *http://www.rvs.uni-bielefeld.de/publications/Incidents/DOCS/ComAndRep/OHare/ NTSB/COPY/ohare-full.html*.
Occupational Safety and Health Administration. "OSHA's Mission. Retrieved May 2006 from: *http://www.osha.gov/*.
Perrow, Charles. 1999. *Normal Accidents*. Princeton University Press.
Robbins, Major Tom. 2002, Nov. New Jackscrew Design Increases Safety Through Redundancy. *Air Safety Weekly 16*, no. 44.

Biographies

Christian G.W. Schnedler is a Corporate Project Manager for DVTel Inc., the market leader in IP Security Solutions. In this capacity, Christian is responsible for enterprise-grade security system deployment projects with duties including managing the integration of complex security systems and developing comprehensive risk mitigation strategies. In addition, Christian is a member of both the American Society for Industrial Security (ASIS) and International Council on Systems Engineering (INCOSE). Christian is currently pursuing a M.E. in Systems Engineering from the Systems Engineering and Engineering Management Department of Stevens Institute of Technology. His research interests are primarily focused on the management of complex systems in the high-technology sector.

Daniel Murphy is a superintendent for Skanska Koch, specialty steel erectors focused primarily on bridge rehabilitation projects in the New York area. Past projects include a NJ Turnpike bridge lowering and total replacement of the lower roadway of the Manhattan Bridge. Currently he is working at the new Yankee Stadium, and is responsible for the fabrication, delivery and erection of the structural precast concrete on the job. Daniel is currently pursuing a M.E. in Engineering Management from Stevens Institute of Technology.

Steven J. Stumpp is a Mechanical Engineer in the Ship Systems Integration and Design Department at the Naval Surface Warfare Center, Carderock Division of the U.S. Navy. Steven has earned a Bachelors Degree in Mechanical Engineering and a Masters Degree in Systems Engineering from Stevens Institute of Technology.

Frantz St. Phar is an electrical engineer for Consolidated Edison of New York Inc. (Con Edison), one of the highest ranked utilities that transmits and distributes gas, electricity, and steam in an economical, reliable, and safe manner. Frantz is responsible for the maintainability of the borough of Manhattan electric distribution system by providing feasible, reliable, and sustainable electric designs. In addition, Frantz provides engineering support for the operators of the Manhattan electric distribution system all year around. Frantz received his B.S. in Electrical Engineering from New York Institute of Technology and his M.E. in Systems Engineering from Stevens Institute of Technology. In addition, Frantz successfully completed the intensive Siemens Power Technology International (P.T.I.) certificate course in Distribution System Engineering and is pursing to obtaining his professional engineering license.

The National Centers for System of Systems Engineering: A Case Study on Shifting the Paradigm for System of Systems

Samuel F. Kovacic, Old Dominion University
Andres Sousa-Poza, Old Dominion University
Charles Keating, Old Dominion University

Introduction

Based on literature and interest encountered in practice, there is a clear and consistent increase in System of Systems Engineering (SOSE). SOSE is being used in organizations such as Lockheed Martin, Concurrent Technologies Corporation, and the Joint Forces Command, all providing a service or solution for complex, distributed problems. However, a quick scan of literature, publications, and websites show that the approaches being presented do not necessarily subscribe to a common theme or methodology for solving, or defining SOS type problems. For example:

- Lockheed Martin (CIAD 2006) proposes a unified discipline methodology that integrates the practitioners needed to provide SOS solutions.
- Concurrent Technologies Corporation (www.sosece.org), that operates the System of Systems Center of Excellence, suggests that System of Systems Engineering is an emergent condition of Systems Engineering and builds on current techniques.
- Joint Forces Command, Standing Joint Task Force (MECS report 2006) focus on System of System Analysis, which is used to define service capabilities in the process of providing for a Joint Force capability.

Continued searches of other institutes and agencies further show no consistent theme for practicing SOSE. A common criticism of SOSE that is beginning to emerge of this nascent field is that the SOSE is neither precisely defined nor adequately distinguished from Systems Engineering. This is a theme that resonates strongly within these organizations and their methods, and possibly an indicator of the cause for such diverse approaches.

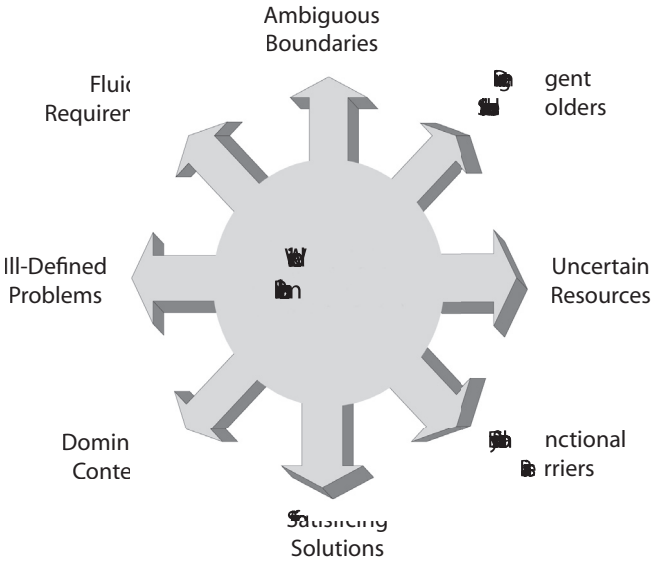
An initial attempt at describing the path for a methodology by the National Centers for System of Systems Engineering (NCSOSE) was published (Keating et al. 2003) titled "Towards A Methodology for System of Systems Engineering." Although cited numerous times as a method for engineering SOS, the paper does not fully document a methodology. Although the paper stops short of providing a definitive solution or approach by which SOS problems can be addressed, it does lay the foundation for transforming knowledge; which is arguably where the document has its greatest value for practice and research. The paper recognizes the dynamic nature of research as well as the energetic value of practice in the field of SOS. Simply stated, transformation can best be observed while change is occurring in practice, and transformation can be compared to existing theory as well as new theory.

The Keating (2003) paper is constantly under scrutiny and assessment by NCSOSE researchers mostly through application of theory in practice. Additionally, as our understanding increases, our perspective of SOSE transforms. This research summarizes the lessons learned from a NCSOSE project and provides the catalyst to a potential new paradigm for SOSE: Complex Situations, a new worldview look at SOS.

NCSOSE

This is a case study of a project conducted by NCSOSE on an agency suffering from the complexities of a wicked problem (Kovacic, Sousa-Poza, and Keating 2006); problems associated with the characteristics in Figure 1.

Figure 1. Wicked Problems



The case study does not focus specifically on the agencies or the problem; instead it focuses on the effectiveness and value of the unique SOSE approach undertaken in this research by NCSOSE.

NCSOSE is a university research center at Old Dominion University established to draw together academic, governmental, and industrial organizations to resolve problems, develop technologies, and direct research concerning major issues in the design, analysis, and transformation of complex systems of systems. Established in October 2002 under congressional auspices, NCSOSE was born out of a recognized need to effectively develop, coordinate, and integrate research and applications to engineer increasingly complex systems that must function as integrated systems of systems. Although there is one unique center titled NCSOSE, it was established with "centers" in the title to accentuate the need to assume a leadership role to foster collaborative efforts in the vision of bringing other entities together through a research network dedicated to issues and applications concerning SOSE. NCSOSE's primary objective is to advance the body of knowledge and state-of-the-art where engineering of complex systems of systems is concerned. NCSOSE supports the development of practical solutions and directs applied research that addresses contemporary SOSE problems; it provides high-quality information resources for those who make decisions, influence policy, and are charged with integration of complex systems of systems; and provides training and education in systems of systems engineering. NCSOSE frequently works in partnership with other research organizations and higher education institutions to enhance the quality of research in systems of systems engineering by promoting the interchange of academic research and knowledge.

NCSOSE, for the purpose of the project, was applying SOSE theory and techniques to a wicked problem. As shown in this case study, SOSE, as defined by NCSOSE, underwent a paradigmatic shift. The remaining chapter describes this transformation of SOSE through the lessons learned derived from the project and the best practices that were undertaken or postulated, and culminates in providing new insights that point to a potential new paradigm that may enable decision makers to deal with complex SoS problems.

The Case Study Design

System of Systems Engineering (Yin 1994) has been suggested as a means to produce successful problem solving and transformation in environments characterized by complex social and technical attributes. The development of methods, models, and environments (simulation based) to permit organizations to more effectively prepare for operation in a complex dynamic environment is a central focus for NCSOSE and essential to SOSE. The constraints researchers and practitioners face are the demand for techniques, methods and models, however, to-date the discipline has not produced a comprehensive toolkit to enable practitioners to effectively perform SOSE. The case study outlined in this paper takes the approach that research must occur in conjunction with practice. The project recognized the dilemma created by performing research and affecting practices in unison and tempered the outcome of the project with a demonstration of capability vice a problem specific solution.

The research aspect of the project was to further the knowledge of the engineering methods associated with "System of Systems" where the practical aspect of the project was to test the mettle of this knowledge with a practical application using innovative concepts and tools. This case study is a summary of the lessons learned and best practices discovered from the project. The project proposal that initiated the contract describes the execution of the project and establishes the foundation for the lesson learned. It is anticipated that the efficacy of the lessons learned and best practices be realized in subsequent projects.

A single exploratory case study approach was used (Tellis 1997) to draw lessons learned from the project that correlate directly to the NCSOSE methodology and indirectly to research currently underway at NCSOSE. The tenets of exploratory research provided the protocol used for the interviews. The coding for the case study addressed the specific items from the following methodology:

- *SOSE philosophy to capture the different level of thinking inherent in the system of systems approach,*
  - *Methodologies that provide guidance and direction for structuring and achieving SOSE initiatives,*
  - *Processes that provide methods for specific aspects of SOSE, and*
  - *Techniques that enhance knowledge and advance practice through specific tools to support SOSE efforts.*
- *SOSE represents an evolution of traditional systems engineering, not a radical departure.*
- *Traditional delineations between research and practice will blur in developing the body of knowledge, methodologies, processes, and techniques to achieve effective SOSE.*
- *SOSE research and products for practice (derived through applications) must include:*
- *From application of SOSE initiatives, best practices must be captured*

Implications for SOSE Methodology

Each of these implications will be assessed in the lessons learned. The conclusion reflects the changes to the

methodology and above points based on the observations and analysis of the project and subsequent case study.

Interviews were conducted with the five major principals responsible for executing the project. The data gathered from the interviews became an important part of the information from which the conclusions of this paper are derived. The semi-formal interviews were intended for exploration of lessons learned and best practices observed over the term of the project. Each interview was conducted by two investigators to cross-reference the results of the interviewees examining the same phenomenon (Denzin 1984).

A face validation technique was used to ensure that the information collected during the interviews had been accurately collected and compiled.

Thematic Analysis (Aronson 1994) was used for data analysis. The principals of the major project elements; project planning, SOSE Environment, SOSE modeling and simulation, and SOS analysis were interviewed. The interviews were based on the project materials provided, literature, and passive observations throughout the project (collected mainly at project delivery). The themes were compiled into a common set of lessons learned and best practices. The lessons learned and best practices were compared to the SOSE implications described earlier.

The themes were extracted from the problems presented. Literature was used to provide an initial structure for the characteristics and sub-characteristics of the case study. As more information was gathered, this structure was expected to either be validated or evolve over the course of the study. Ultimately, the goal is to have the thematic commonalities be based purely on the description of pragmatic problems. With more information or data, this will be achieved through a continuous, inductive research approach utilizing content analysis, and information and dialog mapping techniques.

Information and data were gathered from

1. The project reports including:
    a. Technical proposals
    b. Technical reports
    c. Deliverables
    d. Monthly status reports
2. Open sources such as the internet, including
    a. Information that was collected about the agencies involved in the effort
    b. Information about the problems presented
3. NCSOSE Research
    a. Passive observation
    b. Focused research groups
4. Interviews
    a. Element Leads
    b. Development Leads

The information and data was compiled into a relational structure reflecting the dominant themes articulated in the interview. The structure was influenced strongly by information derived from literature and observation.

The Project

Security, even in its simplest form can be extremely complex due to the irreducibility of the construct. Increasing the granularity through decomposition and adding dimensions rapidly turns security into an intractable problem that defies traditional analytical techniques. Identifying security implications was the project that NCSOSE was assigned, responsible for ferreting out (amidst the myriad of stakeholders, policies, temporal and spatial issues) techniques, processes, models and applications that would accommodate understanding in a situation where complexity increases as detail is added. NCSOSE researchers addressed the project in two venues: research, and application, a lock-step approach was adopted so that synergy between the two venues could be exploited. The intent of the research project was to add to the SOSE body of knowledge. The intent of the application was to create and test methods, tools, and techniques that assist with engineering or managing a SOSE effort.

The objectives were to:

- Develop a model that advances the state of the art with respect to engineering of complex systems of systems.
- Explore the efficacy of the model through application to a specific complex system scenario.
- Investigate the potential for a simulation based approach for deployment of the model to support engineering environments and training.

The scope of this effort includes the research and development leading to an applied model and conceptual design for a simulated environment to facilitate System of Systems Engineering. The System of Systems Engineering model would be applied to a specific security scenario related to the customer's environment. The effectiveness of the model and conceptual design for a simulated environment will be established through assessment of the efficacy of the model in relation to current approaches being used for the system of systems engineering activities for security.

Executing the Project Under SOSE

The project was broken into four major elements: project planning, SOSE environment, SOSE modeling and simulation, and SOS analysis. The following section synopsizes each element, their relationships, and the limitations that emerged from each element during the execution of the project.

Table 1. A Summary of the Tasks Associated with the Project

Program Element	Major Tasks
1. Model and the supporting methodology for SOSE that is appropriate for application to a problem domain for security.	1.1 Survey SOSE research, processes, and models 1.2 Develop SOSE Model 1.3 Develop SOSE Methodology 1.4 Link SOSE Model & Methodology
2. Apply SOSE to frame a specific wicked problem faced by the sponsoring agency.	2.1 Selection of Focus Application System 2.2 Application of SOSE Model and Methodology
2.3 Assessment of SOSE Applicability	
3. Articulate the conceptual design for a simulation-based SOSE environment that is capable of guiding security efforts to deal with complex system problems inherent to the problem domain.	3.1 Conceptual Definition of SOSE Environment 3.2 Investigate Existing Simulation-based Environments 3.3 Establish the Utility of SOSE Environment

Planning

The role for Planning in the project was three fold: obtain and manage funding, interface between external and internal project members, and manage expectations. The proposal was a blend of current management practices for scheduling coupled with a scholarly method for conducting work. The intent was to provide the three operational elements sufficient autonomy to undertake research and application while maintaining accountability for program reviews. Due to emergent conditions and their implications for the project, a significant amount of "Boundary Spanning"—translating internal and external expectations was necessary. Additionally, the "run as you train" philosophy required that the element principal maintain the NCSOSE research role and project problem space with the team at the same time as educate the customer on capabilities and progress. This provided for the distinction of foreground and background activities to act as a "shock absorber" to manage transitions between tasks. The constraints of this approach were:

- The philosophical, methodological, and approach for SOSE was being "born" as the application was progressing
- Access to data and external resources
- Initially, SOSE was unable to provide a clear direction given the nascent nature of the discipline, which also made it difficult to articulate knowledge for the tasks that the elements were to execute.

Offsetting these limitations was the awareness of certain SOSE tenants that generated acceptance of the conditions of emergence, ambiguity and dynamics in a complex system. This awareness would eventually point towards potential new tenants that are addressed at the end of this study.

Environment

The outcome of the task for identifying a SOSE environment evolved over the course of the project. At the beginning, a report describing methods and architectures that would increase SOS understanding was soon superseded by a demonstration that would be necessary to highlight the capability that was intended by this SOSE tool. The agenda for the demonstration was designed to emphasize the capability of this SOSE tool to capture the SOS context.

Modeling and Simulation (M&S)

The role of modeling and simulation was to describe the system versus solving a problem. The contribution of modeling and simulation to the project was execution of data collection strategy, and defining the role modeling and simulation would play in the project as a suitable approach for describing a complex system. This task was extremely complex given that no there was little clarity on what exactly constituted a SOSE methodology for modeling and simulation. It was also unclear in the beginning how modeling and simulation would link to the SOSE research being undertaken due to the need to have multiple teams execute simultaneously without well defined interfaces. The lack of detailed plan, no clear definition of SOSE principles for practice, and data resource constraints were all limitations to the modeling and simulation element of

the problem. This was complicated by time constraints that did not allow a thorough evaluation of best technique to employ for an optimal result, which was exacerbated by limited data access. Regardless of the limitations it was felt that modeling was a reasonable technique for representing a SOS. Modeling and Simulation approaches similar to traditional approaches were employed, enhanced by integration and awareness of context and stakeholder perception making for a more robust model. Methodology perspectives were conceptual and contributed to how a system is framed, realizing a level of understanding that might not have been realized using traditional methods, however it is questionable if this is enough to truly sufficient for positively affecting the transformation of a complex situation.

Analysis

The role of the analysis element was to provide for unified data collection for all elements. Initially viewed as the traditional Systems Engineering aspect of the project it became apparent towards the end that the capabilities provided by the Stakeholder Analysis model, and Initiative mapping were "seeds" to the environment and were a major contribution to the SOS effort. This partially explains the uncertainty that the team members of this element dealt with throughout the project. Even so, the techniques developed by this team would eventually become the preferred method to enhance dialog in a facilitated environment.

Lack of direction, no clear definition of SOSE principles for practice, and data resource constraints were a constant constraint for this element. The "train while running" philosophy compounded the uncertainty experience by this team that had no alternative methods on which they could fall back. The autonomy within the elements, however, facilitated creative development.

SOSE Lesson Learned

Provided in this section is a compilation of the lessons learned derived from the analysis of the interviews conducted with the principle leads of the project. They provide the insights into how the NCSOSE research has been influenced and is transforming based on the increased knowledge accrued from the effort. These lessons learned were the catalyst for NCSOSE to explore a potential new paradigm: complex situation, which provided for new theories to explain the phenomena that occur in social and technical systems that are required to operate within a situation that may be both spatially complex and temporally dynamic.

SOSE Lessons Learned

- ***The degree of emergence in a SOS does not correlate to acquisition and PM activities.*** To expect to map the activities of an extremely emergent system with the foundation of program management, or the structure of requirement and capability engineering is extremely fallible and may cause severe oscillation of project outcomes.
- ***Insufficient philosophical alignment limits management of SOS projects (both internal and external to the project)***. A SOS, such as the one studied in this project, cannot be fully understood due to its complexity and magnitude. This is exacerbated by differing perspectives. Integration of the perspectives is problematic.
- ***Managing expectations are different than "Design of Forum***." There is a proactive and reactive component to expectation management. Combining the two will only convolute the attempt of managing and structuring events.
- ***Research and applications need to be purposefully designed and remain flexible as the project progresses***. To *design* a SOS; a complex situation needs to be static long enough to enact a long term plan and then expect to follow the plan strictly for most SOS problems, is somewhat naïve. The two tasks (planning and design) are diametrically opposed.
- ***SOSE outputs are deliverables, however, more important are outcomes which may or may not be tangible (i.e., understanding, systemic inquiry).***
- ***Difficulty in balancing "holistic" framing of SoS versus the need to conduct analysis—at some level the system is irreducible."*** It is difficult to judge where irreducibility has been achieved in an analytic process. Paradoxically an impossible task, particularly within multiple perspectives
- ***Every SOS application has a research component and application component. There are different mindsets of what constitutes success in each worldview; success in SOS requires balancing multiple worldviews, and determining where movement forward can be achieved and where movement forward cannot be achieved.***
- ***Managing the balance. Integration of SOS needs to be a continuum, to introduce vehicles to facilitate communication and coordination due to emergence.***
- ***Planning can't occur before framing.*** It is important to allow for sufficient discovery of the problem domain (not just through literature and interviews, but site visits and tours as well) before beginning the planning phase. A pre-site survey is an excellent tool for facilitating framing. *This approach assumes an alternate approach for a solution for planning in a SOS. An alternate view is that planning is extremely fallible in complex situation (a tenant of complex situations).*

- ***Integration between teams is difficult—don't expect cohesive communication to occur when dealing with distinctly separate teams—cross pollination must occur.***
- ***Objective of modeling is to capture large systems with unclear inter-relationships.*** Modeling and simulation allows perspectives to be represented in a SOS. Modeling and simulation is necessary for unifying multiple perspectives into a holistic perspective. *This perspective assumes integration of multiple perspectives can be achieved. An alternate view is that integration of perspectives is extremely fallible in complex situation (a tenant of complex situations).*
- ***Traditional Techniques can be mixed with conceptual statements. Although mixing was necessary, this may still not be sufficient to provide a solution. Verifying conceptual framework is crucial to SOSE for movement forward.***
- ***The complexity of SOS does not lend itself to clear and precise framing or representing of the problem or system in question***. There is no clear solution in literature, research, or practice on how best to bound a wicked problem, this fundamental failing is a pivotal aspect in a SOS project. *This forms the basis for the argument that a SOS can never be engineered in the traditional sense (a tenant of complex situations).*
- ***Knowledge and understanding is a symbiotic relationship between the developers and practitioners that occur over the span of the project. This is in contrast to full understanding experienced at the beginning of a project (typical of other traditional or mainstream methods).***
- ***Models, although they provide structure, are not representative of the world and should be viewed as tool for dialog rather than for decision or solution***. There was a lot of emphasis on the model (particularly from the consumer) as a measure of success in the program this resisted or blocked the other key elements of the project.

SOSE Best Practices

Flexibility allowed for a demonstration to emerge as an unstated requirement for integration of domains. Similarly, flexibility allowed roles and task to shift based on new understanding and the ever changing nature of SOS. Given the lack of maturity of a discipline, a high dependency on people to "do the right things" with minimal guidance and oversight can be a double-edged sword—sometimes it works, sometimes it doesn't, however it is very conducive for exploratory research. A clear and concise methodology provides a foundation for overcoming the inadequacies of resources and assets. Coordination between components enables project success providing excellent cross pollination of ideas and checks and balances for keeping the team (or teams) together.

System Dynamics based modeling is appropriate for large complex systems and SOS at macro level while multiple team interactions induce creativity. Gaming is also seen as a productive method to generate creativity. A facilitated dialog is crucial for integration. Modeling as a learning technique enabled dialogs to revolve around understanding and potential solution sets. Dynamic Group Planning was an effective tool for overcoming the emergent conditions experienced in trying to force structure on a dynamic complex system.

Conclusion—Waiving the Magic Wand—Complex Situations

Endemic to this case study was the thematic implication that SOS was a phenomenon of a more prevalent condition: Complex Situations. Analysis was conducted on the responses from interviews in terms of how the research affected the theoretical constructs that evolved from the project in terms of Complex Situations.

Theme 1

The distinction that is made between situations and systems lies primarily in the recognition of discordant worldviews. (MECS report 2006)

It is critical to gauge compatibility of paradigms between those requesting versus those providing. Overall success of the project is affected by incompatible worldviews affecting all phases of program and acquisition life cycle. Without the correct worldview alignment movement forward will never occur. Diagnostics for understanding complex situations versus the SOS will help align worldviews or accommodate worldviews and permit clearer exchange of ideas/progress for a better integrated effort.

Theme 2

A simple situation is a situation in which the level of understanding that an observer(s) has is relatively high at any point in time and knowledge claims are bound to have a high probability of being correct. (MECS report 2006)

Even simple situations have challenges, the distinction is not a lack of complexity but a high level of understanding; this requires a constant balancing of the domain to assure alignment of domains. A suggested approach for a simple complex problem framing may be:

- Research root of problem
- Accurately bound based on goal
- Utilize models to generate the dialog to build understanding
- Develop actions to move toward goals.

Theme 3

*A complex situation is a situation in which, for any number of reasons, the level of understanding that an observer(s) has of the situation is extremely low at any point in time, and knowledge claims are bound to have a high probability of being erroneous. (*MECS report 2006)

Bound the application early in the project and use the application itself to refine the methodology. It was unanimous that bounding the problem in a complex situation is critical; however, every attempt at bounding only highlighted the challenges of building a coherent reducible domain. A diagnostic capability is being explored by NCSOSE that reframes how to view the picture, within a framework, that can maintain the integrity of all domains from inception to transformation.

A NCSOSE core capability is SOS research; in the course of executing the NCSOSE mission we have discovered a core paradigm shift to a broader issue: complex situations. Research into this new paradigm will be redirected to SOS with the expected result that the new worldview will provide for insights into overcoming the barriers plaguing SOSE.

References

Aronson, Jodi. 1994. A pragmatic view of thematic analysis. *The Qualitative Report 2*, no. 1.

Denzin, N. 1984. *The research act.* Englewood Cliffs, NJ: Prentice Hall.

Joint Forces Command Problem Analysis. 2006. *MECS Summit Report 1,* no. 13.

Keating, Charles et al. 2003. Towards a methodology for system of systems engineering.

Kovacic Samuel F., Andres Sousa-Poza, and Charles Keating. 2006. Complex situations: An alternative approach for viewing a system of systems.

Tellis**,** Winston. 1997, July. Introduction to Case Study. *The Qualitative Report 3*, no. 2.

Yin, R. 1994. *Case study research: Design and methods* (2nd ed.). Beverly Hills, CA: Sage Publishing.

Biographies

Samuel F Kovacic is a Research Scientist with the National Centers for System of Systems Engineering (NCSOSE) at Old Dominion University. He is currently pursuing a Ph.D. in Engineering Management from Old Dominion University, holds a MBA in Aviation from Embry Riddle Aeronautical University, and a BS.c. in Computer Science from the University of Maryland. His primary research is in the philosophical constructs of worldviews and the study of management and engineering practice as it pertains to complex situations, system of systems, strategic management and organizational change. He retired from active duty with over 22 years service in the United States Air Force.

Andres Sousa-Poza, Ph.D., is an Assistant Professor in the department of Engineering Management and Systems Engineering at Old Dominion University. He obtained a Ph.D. and a M.S. in Engineering Management from the University of Missouri-Rolla, and a BS.c. in Mechanical Engineering from the University of Cape Town, South Africa. His primary research interests lie in the study of management and engineering practice as it pertains to complex situations, system of systems, strategic management and organizational change. He has worked, studied and lived in North and South America, Europe, and Southern Africa.

Charles Keating, PhD, is a Professor in the department of engineering management and systems engineering at Old Dominion University. He received a B.S. in Engineering from West Point, an M.A. in Management from Central Michigan University, and a Ph.D. in Engineering Management from Old Dominion University.

Guide for Authors

The *Systems Research Forum* will accept submissions in English that will be peer reviewed for potential acceptance and publication in the journal. Papers will be evaluated on:

- Relation to the field of systems engineering
- Advancement in the state of knowledge of the field
- Quality of scholarly presentation and investigative rigor

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Final selection of papers for publication, and the form of publication, shall rest with the Editors-in-Chief. The review process is estimated to take three to five months.

Submission Instructions

A cover letter must accompany each submission indicating the name, address, telephone number, fax number, and e-mail address of the author to whom all correspondence is to be addressed. An affiliation must be supplied for each author. If the manuscript has been presented, published, or submitted for publication elsewhere, please inform the Editors-in-Chief. Our primary objective is to publish technical material not otherwise available.

Prospective authors should submit electronic copies of the complete manuscript, including tables and illustrations, to:

Dr. Rashmi Jain
Stevens Institute of Technology
School of Systems and Enterprises
Castle Point on Hudson
Hoboken, NJ 07030
Tel: 201.216.8047
rashmi.jain@stevens.edu

Dr. Brian Sauser
Stevens Institute of Technology
School of Systems and Enterprises
Castle Point on Hudson
Hoboken, NJ 07030
Tel: 201.216.8589
brian.sauser@stevens.edu

Format

Manuscripts should contain: Title; names and complete affiliations of authors, including phone number, facsimile number, and e-mail address. Please provide an informative 100 to 200 word nonmathematical abstract at the beginning of your manuscript suitable for retrieval purposes. The abstract should provide an overview of your paper and not a statement of conclusions only. It should not contain literature citations. If applicable, please provide keywords; contract grant sponsor(s); contract grant numbers(s). Papers are normally about 10 to 15 published journal pages in length. Authors should write succinctly and should note that a significant element of the review process will be the paper length relative to its content and the clarity of writing.

For further instruction on styling of paper, see Style Guide below. For case studies, see guidance under Case Study Format on the next page.

Style Guide

Format. Papers must be submitted in English. No maximum length exists, but authors should write concisely. A significant review element will be the paper's length relative to its content. Papers should be double spaced and in single column format. Tables and figures are labeled Table or Figure not Exhibit, numbered consecutively.

Footnotes. Footnotes are strongly discouraged. If unavoidable, they are preferred to endnotes.

References. References should be complete and clear. Please refer to the *Chicago Manual of Style* Author-Date Style for questions regarding format. For all authors, full names are required. For periodicals, include volume #, issue #, month/quarter, year, and page numbers. For book chapters, include editor's full name(s), publisher, and page numbers. Cite each reference in the text by author and year without a separating comma. Only cited references and publications that are readily available should be included.

Reference Examples:

Periodical
Emes, M., Alan Smith, and Douglas Cowper. 2006. Confronting an identity crisis—How to "brand" systems engineering. *Systems Engineering 8*, no. 2.

Book
Ashby, W. R. 1956. *An introduction to cybernetics*. London: Chapman and Hall.

Edited Book
Bijker, W., Thomas P. Hughes, and Trevor Pinch, eds. 1987. *The social construction of technological systems: New directions in the sociology and history of technology*. Cambridge: MIT Press.

Tables/Figures. For all tables/figures, use Arial, 10-pt maximum and place the title above the exhibit. Tables/figures prepared for live presentation (usually 14 pt or larger) MUST be modified before submission. Use portrait layout where possible. Do not box tables/figures. If the paper is accepted, all tables/figures will need to be sent in their native files as separate attachments.

Style. Write clearly, simply, and directly. Use "I" or "we," not "the author(s)." Data should be rounded to 3 or fewer significant digits.

Case Study Format

The headings below represent generic headers. They can be changed to reflect the content of the paper. The basic sections that should be included are:

- Introduction
  - Give the reader background materials to set the stage for the problem statement. This would be considered a theoretical background. Think of it as a foundation to the problem statement.
  - Clarify for the reader the author's understanding of the situation.
  - Statement of the Problem and objectives of study.
- Presentation of the Model/Tools/Process
  - If a model/tool/process was analyzed or used to define the case, it should be explained here.
- Methodology
  - Here the paper should define the problem, what was analyzed, what was done in the study.
  - There should be some explanation on how the study was conducted… methodology, mode of analysis, and sources of data.
- Case Overview
  - Provide a brief over of the case(s), background, and significant events of the project(s).
- Analysis
  - Report on the case analysis and the results as defined by the model/tool/process
- Solutions, Implications, Recommendations
  - What was learned, how does it impact systems engineering.
  - What are the future implications based on the results of this case study.