System Interoperability Influence on System of Systems Engineering Effort

Jo Ann Lane  
University of Southern California  
Los Angeles, CA  
jolane at usc.edu

Ricardo Valerdi  
Massachusetts Institute of Technology  
Cambridge, MA  
rvalerdi at mit.edu

Abstract

An important characteristic of a System of Systems (SoS) is interoperability among its constituent systems. It enables the flow of information and the seamless introduction of new systems into the SoS. But interoperability comes at a price. Current studies indicate that there is significant engineering effort involved in making systems interoperable. However, this feature is not adequately represented in current cost models. To characterize and quantify the interoperability (or non-interoperability) influence on SoS engineering effort, this paper analyzes 14 interoperability models and presents two approaches that can be used as an extension to the COSYSMO or COSYSMO for SoS cost models.

Introduction

System interoperability has been defined in a multitude of ways to understand various aspects and levels of interoperability. (Ford et al. 2007a) identified 34 definitions of system interoperability. However, the most common definition of system interoperability by far according to (Ford et al. 2007a) is the Department of Defense (DoD) 1977 definition: The ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together.

Studying interoperability with respect to systems engineering cost estimation involves three steps of increasing difficulty: characterization, quantification, and estimation. This paper provides a comprehensive characterization of system interoperability, evaluates current models that quantify interoperability at some level, and presents two approaches for quantifying the engineering effort involved with developing system capabilities that require the support of two or more interoperable systems. The steps involved with estimation are left for future work since they require historical project data to validate cost estimating relationships.

Understanding Interoperability and Its Potential Impacts Through Examples

Many systems today depend on interoperability in order to function. Some common examples are:

- **Healthcare**: Providers can share patient information between primary care physicians, specialists, laboratories, pharmacies, and insurance providers through electronic patient records.
- **Television**: Home televisions can access content providers through a variety of methods with the appropriate subscriber equipment (satellite dish, cable box, etc.). In addition,
some providers can also use the same equipment to provide internet and phone access. Note however, that different countries have different standards for interoperability and this provides challenges as people try to use DVDs around the world.

- **Transportation:** Railways have greater or lesser interoperability depending on how well they conform to standards for such things as gauge, couplings, brakes, signalling, communications, loading gauge, and operating rules. North American railroads are highly interoperable while Europe, Asia, Africa, Central and South America, and Australia much less so. The parameter most difficult to overcome (at a reasonable cost) is the incompatibility of gauge, though variable gauge axle systems exist.

- **Communication networks:** A multi-band mobile phone in the U.S. is able to subscribe to domestic networks using Code Division Multiple Access (CDMA) or Time Division Multiple Access (TDMA) technologies while also being able to subscribe to overseas networks using the Global System for Mobile Communications (GSM).

In addition, today many crisis response organizations are incorporating various technologies in their platforms and systems so that they can better interoperate in response to sudden crises, as shown in Figure 1. These capabilities have been very important in fighting recent wild fires in Southern California (CDFFP 2011).

![Figure 1. Regional Area Crisis Management System of Systems.](image)

However, as an example of what can happen when interoperability is missing, when the Cedar Fire occurred in San Diego County in 2003, local and regional responders were severely overwhelmed. In response, state and federal military organizations who were in the area volunteered to help fight the fires—they had both fire-fighting equipment and trained personnel to operate the equipment. However, they ended up on the sidelines watching homes and businesses burn because they could not communicate with the local responders and it was deemed too risky to have them in the fire-fighting areas without communications with the other responders. After the fire was extinguished, county and state officials conducted studies and
made changes to improve interoperability of the local, regional, and federal crises response systems. When the next round of catastrophic fires occurred in 2007, the improved interoperability paid off and considerably more resources were quickly deployed to fight the fires and evacuate people.

Because most systems today need to interoperate with other systems, interoperability assessment and planning is a key part of systems engineering planning. The following section looks at the current state of the practice with respect to system interoperability and systems engineering planning and cost estimation.

**Current System Interoperability and Cost Estimation State of the Practice**

Considerable work has been done to extend the Constructive Systems Engineering Cost Model (*COSYSMO*) to estimate effort associated with the engineering of system of systems (SoS) capabilities (Lane 2009). This *COSYSMO* cost model extension for SoS is based on the SoS engineering (SoSE) processes described in the Department of Defense (DoD) *Systems Engineering Guide for System of Systems* (DoD 2008). Anecdotal evidence from many SoSE projects (NRC 1999; Sledge 2010) indicates that a major challenge in both maintaining existing SoS capabilities and developing new SoS capabilities is constituent system non-interoperability. For net-centric SoS, this interoperability issue is often related the compatibility of interface protocols and internal system data models that manage the data and information shared across the SoS constituent systems (Fisher 2006; Smith et al. 2006). This challenge is exacerbated as systems become part of multiple SoSs because there are inconsistent standards across SoSs.

The commercial sector has turned to Service-Oriented Architectures (SOAs) to deal with this problem for enterprise-wide solutions. This approach improves interoperability, but requires a major upfront investment and continual care and feeding as systems come and go within the SoS enterprise (Fisher 2006; Lewis et al. 2010) and is not always well-suited for certain environments such as applications where one way asynchronous communications are necessary, situations where loose coupling is not desired or even detrimental, or applications requiring an extensive graphical user interface with high levels of data exchange (e.g., map rendering and manipulation) (Exrforys Inc. 2010). Even in those situations where SOAs may be appropriate, the organizations contributing to or “owning” the SoS may need to establish a SOA governance model to oversee SOA design decisions or guide extensive system modifications and refactoring for effective SOA implementation (IBM 2010).

Others (NRC 1999; USAF SAB 2005) have suggested that convergent protocols and data standards are a more efficient long term solution that can establish more of a “plug and play” environment with minimal data and protocol conversions. This is especially important in minimizing processing overhead in situations where high throughput or performance (e.g., response time, precision) is required. SoS SE research continues in this area to develop frameworks to better understand SoS interoperability issues as well as various approaches to support SoS interoperability (Lewis, et al. 2010).

Despite the various approaches that exist to promote system interoperability and enable interoperability in the SoS environment (e.g., the Net-Ready Key Performance Parameter (USD AT&L 2008) and related high level requirements), we lack a method for determining how much additional engineering effort is needed when addressing interoperability issues. These include the ability to:
• Provide acceptable levels of interoperability in environments where systems come and go from an SoS and many single systems operate in multiple SoSs
• Implement new cross-cutting capabilities given the current level of interoperability or non-interoperability
• Conduct tradeoffs between various mitigation approaches.

The following section looks at existing interoperability models that might be useful in assessing “as is” and “to be” system interoperability and the associated effort to transition from “as is” to “to be”.

**Analysis of Existing Interoperability Models**

In order to characterize the amount and type of engineering work needed to adequately meet interoperability goals, we need to identify various types and levels of interoperability that allow systems to interoperate in various environments. A recent survey (Ford et al 2007a) has identified 14 interoperability models. These models are briefly described below.

**Overview of Existing Interoperability Models**

Each of the interoperability models described in (Ford et al. 2007a) is briefly described below with respect to how it models interoperability, its primary usage, and the original source that defined it. These models are shown in chronological order to highlight the evolutionary aspect of this group.

**Spectrum of Interoperability Model (SoIM) (LaVean 1980):** SoIM combines a technical level (1-4) with a management/control level (1-6) to determine an interoperability level ranging from 1 to 7. The interoperability levels are separate systems, shared resources, gateways, multiple entry points, conformable/compatible systems, completely interoperable systems, and same system.

**Quantification of Interoperability Methodology (QoIM) (Mensh et al. 1989):** QoIM quantifies seven interoperability components (media, languages, standards, requirements environment, procedures, and human factors) to evaluate the interoperability of systems, units, or forces with respect to a given mission scenario.

**Military Communications and Information Systems Interoperability (MCISI) (Amanowicz et al. 1996):** MCISI models communications and information system (CIS) interoperability using “level of command”, “CIS services”, and “transmission medium” using red, yellow, or green to indicate none, partial, or full interoperability in a 3D matrix. Further mathematical analysis views systems as points in a multi-dimensional space and calculates an interoperability measure between two features in different systems by computing their “distance” from each other.

**Levels of Information System Interoperability Model (LISI) (DoD 1998):** LISI is similar to the SoIM model. It has five levels (isolated, connected, functional, domain, and enterprise) instead of the SOIM seven levels and adds four attributes within each level. The new attributes included in the model are procedures, applications, infrastructure, and data (PAID). This model is best suited for measuring information systems interoperability. Its outputs include a “highest common level of interoperability between two information systems across all PAID attributes” and a “matrix that can be used to visualize the interoperability of a group of systems”.

**Analysis of Existing Interoperability Models**

In order to characterize the amount and type of engineering work needed to adequately meet interoperability goals, we need to identify various types and levels of interoperability that allow systems to interoperate in various environments. A recent survey (Ford et al 2007a) has identified 14 interoperability models. These models are briefly described below.

**Overview of Existing Interoperability Models**

Each of the interoperability models described in (Ford et al. 2007a) is briefly described below with respect to how it models interoperability, its primary usage, and the original source that defined it. These models are shown in chronological order to highlight the evolutionary aspect of this group.

**Spectrum of Interoperability Model (SoIM) (LaVean 1980):** SoIM combines a technical level (1-4) with a management/control level (1-6) to determine an interoperability level ranging from 1 to 7. The interoperability levels are separate systems, shared resources, gateways, multiple entry points, conformable/compatible systems, completely interoperable systems, and same system.

**Quantification of Interoperability Methodology (QoIM) (Mensh et al. 1989):** QoIM quantifies seven interoperability components (media, languages, standards, requirements environment, procedures, and human factors) to evaluate the interoperability of systems, units, or forces with respect to a given mission scenario.

**Military Communications and Information Systems Interoperability (MCISI) (Amanowicz et al. 1996):** MCISI models communications and information system (CIS) interoperability using “level of command”, “CIS services”, and “transmission medium” using red, yellow, or green to indicate none, partial, or full interoperability in a 3D matrix. Further mathematical analysis views systems as points in a multi-dimensional space and calculates an interoperability measure between two features in different systems by computing their “distance” from each other.

**Levels of Information System Interoperability Model (LISI) (DoD 1998):** LISI is similar to the SoIM model. It has five levels (isolated, connected, functional, domain, and enterprise) instead of the SOIM seven levels and adds four attributes within each level. The new attributes included in the model are procedures, applications, infrastructure, and data (PAID). This model is best suited for measuring information systems interoperability. Its outputs include a “highest common level of interoperability between two information systems across all PAID attributes” and a “matrix that can be used to visualize the interoperability of a group of systems”.
Interoperability Assessment Methodology (IAM) (Leite 1998): IAM is similar to the QoIM model. IAM contains nine interoperability components (requirements, standards, data elements, node connectivity, protocols, information flow, latency, interpretation, and information utilization) as compared to QoIM’s seven components. IAM also includes “degrees of interconnection”: connectivity, availability, interpretation, understanding, utility, execution, and feedback. Some have used IAM to extend the LISI model.

Organizational Interoperability Maturity Model for C2 (OIM) (Clark et al., 1998): OIM is an extension of the LISI model that incorporates organizational aspects of interoperability. It is used to identify organizational problems and evaluate interoperability in specific scenarios for a Command and Control (C2) coalition operation. It has five levels of organizational interoperability (independent, cooperative, collaborative, combined, and unified) and four organizational interoperability attributes (preparation, understanding, command and coordination, and ethos).

Stoplight (Hamilton et al. 2002): Stoplight is a simplified model used to help decision makers determine if legacy systems meet operational and acquisition interoperability requirements. It is implemented as a 2D matrix with “yes/no” responses in the matrix. It can also be used to track interoperability improvements over time.

Levels of Conceptual Interoperability Model (LCIM) (Tolk et al. 2003): LCIM is similar to LISI and OIM, but is used in the early stages (conceptual design) to determine if meaningful interoperability between the systems is possible. The five levels of conceptual interoperability (system-specific data, documented data, aligned static data, aligned dynamic data, harmonized data) were later changed and extended to no interoperability, technical interoperability, syntactic interoperability, semantic interoperability, pragmatic interoperability, dynamic interoperability, and conceptual interoperability.

Layers of Coalition Interoperability (LCI) (Tolk 2003): LCI is a single model that combines technical and operational interoperability using nine layers of interoperability (physical interoperability, protocol interoperability, data/object model interoperability, information interoperability, knowledge/awareness, aligned procedures, aligned operations, harmonized strategy doctrines, and political objectives).

NATO C3 Technical Architecture Reference Model for Interoperability (NMI) (NATO 2003): Version 4 of this model was an update to closely reflect the LISI model. It has four degrees of interoperability (unstructured data exchange, structured data exchange, seamless sharing of data, and seamless sharing of information) which when combined with “no interoperability”, maps onto the LISI five levels of interoperability.

System-of-Systems Interoperability Model (SoSI) (Morris et al. 2004): SoSI was designed to support the Software Engineering Institute’s system of systems interoperability research. The model is based on three levels of interoperability (operational, constructional, and programmatic) and the activities associated with each. This model lacks specific metrics to quantify interoperability within a system of systems.

Non-Technical Interoperability Framework (NTI) (Stewart, et al. 2004): NTI is based on the OIM organizational model. The four OIM attributes are the core of this model. However, this model provides a more detailed breakdown of these attributes. NTI also extends the OIM to calculate a “Multinational Forces Co-operability Index” that provides a quantitative assessment (from 1 to 16) of the attributes “preparation” and “understanding”. 

Organizational Interoperability Agility Model (OIAM) (Kingston et al. 2005): OIAM builds upon the OIM organizational model. This extension is designed to capture the agility of
systems to form and reform into coalitions/systems of systems and to adapt to changing circumstances. OIAM redefines the OIM five levels as static, amenable, accommodating, open, and dynamic. OIAM also uses the four OIM attributes, but combines preparation and understanding into a single attribute.

**The Layered Interoperability Score (i-Score) (Ford et al. 2007b):** i-Score is a mathematical method designed to measure the interoperability of all types of systems (e.g., technical, biological, organizational, environmental) for a specific operational scenario/thread. The method is based upon a matrix that captures the number of systems supporting the thread, the number of times the system is used in the thread, and the level of interoperability between each interacting pair (no human or machine translation needed for a system pair, machine translation required, or human translation required).

**Comparative Analysis of Interoperability Models**

For the most part, these models focus on interoperability of information systems, i.e., information/data sharing over external interfaces. A few also incorporate organizational aspects of interoperability. Of key note is the relationships between some of these models: several models are refinements or extensions of earlier models.

It is also interesting to note that while interoperability seems to be a programmatic concern as more systems are integrated into systems of systems, Ford’s research (Ford et al, 2007a) did not provide much evidence that any of these models had been institutionalized to any great extent either within development organizations or government system acquisition organizations. The only clear example was the LISI model that was referenced in the DoD CJCSI 6212.01C instruction on interoperability, but later dropped in the next version (CJCSI 6212.01D).

**Synthesis of Interoperability Models into COSYSMO Framework**

The first step in the synthesis of the system interoperability models is determine the subset of models that are 1) mature enough and 2) can support an assessment of interoperability with respect to cost estimation. As noted above, many of the later system interoperability models build or elaborate on earlier models. In addition, the models evaluated focused on “technical” interoperability of the systems as well as “organizational” interoperability. While organizational interoperability is important, it is not directly related to the activities for which COSYSMO estimates effort. Therefore, it appears that the best models for evaluating interoperability as it affects systems engineering effort are LISI for the development of new capabilities within a set of existing systems and LCIM for systems that are still in the conceptual phase of development.

The next step is to determine how to incorporate the model levels of interoperability into the COSYSMO cost model. The basic form of the COSYSMO model is an engineering size estimate raised to a calibrated exponent (to account for the diseconomy of scale as size grows (Valerdi et al. 2011)) multiplied by an effort adjustment factor (which is the product of a set of effort factors). So the options are to incorporate the interoperability factor as an additional effort adjustment factor or to use it to compute the system size parameter. The actual approach will depend on the analysis of additional systems engineering data. If the level of system interoperability affects systems engineering effort in a linear way, it should be incorporated as an additional effort adjustment factor. If the level of system interoperability affects systems engineering effort in an exponential way as system size grows (or as the number of interfaces
grows), it should be incorporated as an interface size weight. Each of these approaches is presented below.

**Method 1: Interoperability Effort Adjustment Factor**

Table 2 shows how the assessed level of interoperability can be used to define a new effort adjustment factor to be incorporated into the COSYSMO cost model. Two ways to apply a level of interoperability are provided: one to evaluate interoperability in a set of existing systems and another to evaluate interoperability in the conceptual development phase for a new system.

**Table 2: Level of SoS Interoperability**

<table>
<thead>
<tr>
<th>Type of Development</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing systems (based upon LISI levels)</td>
<td>Isolated</td>
<td>Connected</td>
<td>Functional standards employed</td>
<td>Domain standards employed</td>
<td>Enterprise standards employed</td>
</tr>
<tr>
<td>New system(s) (based upon LCIM conceptual levels)</td>
<td>System-specific data</td>
<td>Documented data</td>
<td>Aligned static data</td>
<td>Aligned dynamic data</td>
<td>Harmonized data</td>
</tr>
</tbody>
</table>

The assumption is that High and Very High values in the interoperability factor are associated with a cost savings while Low and Very Low values are a cost penalty. The actual values need to be determined through analysis of historical data.

**Method 2: Interoperability Assessment as Part of Interface Size Driver**

As mentioned early in the paper, interface interoperability can be difficult to manage as systems come and go in various systems of systems. In some systems the interfaces are the most important drivers of systems engineering effort. While it is possible for the greatest leverage in system architecting to be at the interfaces (Rechtin 1991), this leverage may come at a significant cost. Both the quantity and complexity of interfaces often lead to more systems engineering effort. The operational definition of the number system interfaces driver is provided below with the corresponding rating scale in Table 3. The last row in Table 3 has been added to reflect the impact of interoperability on the complexity of system interfaces.

**Table 3: Number of System Interfaces Rating Scale**

<table>
<thead>
<tr>
<th>Easy</th>
<th>Medium</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple messages and protocols</td>
<td>Moderate communication complexity</td>
<td>Complex protocol(s)</td>
</tr>
<tr>
<td>Uncoupled</td>
<td>Loosely coupled</td>
<td>Tightly coupled</td>
</tr>
<tr>
<td>Strong consensus among stakeholders</td>
<td>Moderate consensus among stakeholders</td>
<td>Low consensus among stakeholders</td>
</tr>
<tr>
<td>Well behaved</td>
<td>Predictable behavior</td>
<td>Emergent behavior</td>
</tr>
<tr>
<td><strong>Domain or enterprise standards employed</strong></td>
<td><strong>Functional standards employed</strong></td>
<td><strong>Isolated or connected systems with few or no standards</strong></td>
</tr>
</tbody>
</table>
This driver represents the number of shared physical and logical boundaries between system components or functions (internal interfaces) and those external to the system (external interfaces). These interfaces typically can be quantified by counting the number of external and internal system interfaces among the system elements. The assumption is that there is an increasing level of cost associated with Easy, Medium, and Difficult levels of interoperability. The actual values need to be determined through analysis of historical data.

**Implications for Researchers and Practitioners**

This work has a number of important implications for researchers. First, when studying interoperability it is necessary to look at the broader view that involves not just technology but also governance, operating procedures, training, and usage. This broader view is part of making interoperability a feasible capability. While important for success, the effort for these activities is not covered in models such as COSYSMO. Second, quantifying interoperability is a necessary step in the evolution of our understanding of the factors that affect SoSE effort. Overlooking the impact of interoperability on project effort can cause severe delays and cost overruns downstream.

For practitioners there are important implications in the planning and execution of interoperability requirements in SoS. By applying the concepts presented here, planners can better forecast the complexity and cost involved with acquiring interoperable systems. Moreover, the concepts presented here can help identify interoperability related risks that could influence the feasibility of a project.

**Conclusions and Next Steps**

The importance of interoperability is evident across a variety of systems; from healthcare, transportation, and communications. Characterizing the degree of interoperability needed in systems of systems in an important aspect of good resource planning. Beyond characterizing this system attribute, it is important to quantify the levels of interoperability difficulty using rating scales. Eventually the objective is to incorporate interoperability considerations into cost models so that planners can accurately forecast its impact on project execution.

At this point in time, researchers and practitioners can both benefit from this work since it provides a method for thinking about interoperability and its role in project complexity. The next step in this process is to collect additional systems engineering effort data to evaluate both methods presented above for incorporating interoperability into the COSYSMO model. The goal of this step is to determine which approach better reflects the influence of interoperability and an appropriate set of scale factors for Method 1 or an appropriate set of size driver weights for Method 2.

**References**


**Biography**

**Dr. Jo Ann Lane** is a research assistant professor at the University of Southern California Center for Systems and Software Engineering, conducting research in the areas of SoSE, systems engineering, and innovation. She was a co-author of the 2008 Department of Defense *Systems Engineering Guide for Systems of Systems*. She received her PhD in systems engineering from the USC and her Master’s in computer science from San Diego State University.

**Dr. Ricardo Valerdi** is a Research Associate at the Massachusetts Institute of Technology. He received his BS/BA in electrical engineering from the University of San Diego in 1999, and his MS and PhD degrees in systems architecting and engineering from USC in 2002 and 2005. He served as program chair for the 20th and 24th Forums on COCOMO and Software Cost Modeling and on the INCOSE Board of Directors. He is the co-Editor-in-Chief of the *Journal of Enterprise Transformation*.