Integration maturity metrics: Development of an integration readiness level, SSE_S&EM_004_2007

Article · January 2010

4 authors, including:

Brian Sauser
University of North Texas

Information Knowledge Systems Management 9 (2010) 17–46
DOI 10.3233/IKS-2010-0133
IOS Press

Integration maturity metrics: Development of an integration readiness level

Brian Sauser[a], Ryan Gove[a], Eric Forbes[b] and Jose Emmanuel Ramirez-Marquez[a]
[a]*Stevens Institute of Technology, School of Systems and Enterprises, Systems Development & Maturity Laboratory, Castle Point on Hudson, Hoboken, NJ, USA*
Tel.: +1 201 216 8589; E-mail: bsauser@stevens.edu
[b]*Northrop Grumman Corporation, Mission Systems Sector, 300 M Street SE, Washington, DC, USA*

Abstract: In order to optimize the process of complex system integration, it is necessary to first improve the management of the process. This can be accomplished through the use of a generally understood metric. One such metric is Technology Readiness Level (TRL), which is used to determine technology maturity, but does not address integration maturity. Integration Maturity Metric (IMM) requirements are developed through review of aerospace and defense related literature. These requirements are applied to currently existing integration maturity metrics, and the proposed Integration Readiness Level (IRL). IRL is then refined to fully meet these requirements, and applied to three aerospace case studies, along with the other identified metrics, to compare and contrast the results obtained.

Keywords: Technology readiness level, integration readiness level, integration

1. Introduction

Buede [5] defines system integration as “the process of assembling the system from its components, which must be assembled from their configuration items.” By this definition, system integration could intuitively be interpreted as a simplistic process of “putting together” a system from its components, which in-turn are built from configuration items. However, as Buede later explains, integration is a complex process containing multiple overlapping and iterative tasks meant to not only “put together” the system but create a successful system built to user requirements that can function in the environment it was intended for.

This simple yet effective definition implies a structure to system integration, referred to as simply integration from this point forward. This structure is often described in the systems engineering (SE) process as being the “upward slope” of the traditional V-model (see Fig. 1) [5]. It starts with configuration item integration and ends with verification and validation of the complete system in the operational environment. Moving from simply integrating configuration items to integrating the system into its relevant environment is a significant effort that requires not only disciplined engineering, but also effective management of the entire SE process. While disciplined engineering is something that can be achieved through the use of mathematics and physics, effective management of the SE process is a much less structured and quantitative activity. In fact, there is no one standard methodology to follow when considering the integration of most systems. This issue becomes magnified as the complexity of system design and scope increases, implying the need for a method to manage the integration process [4]. The

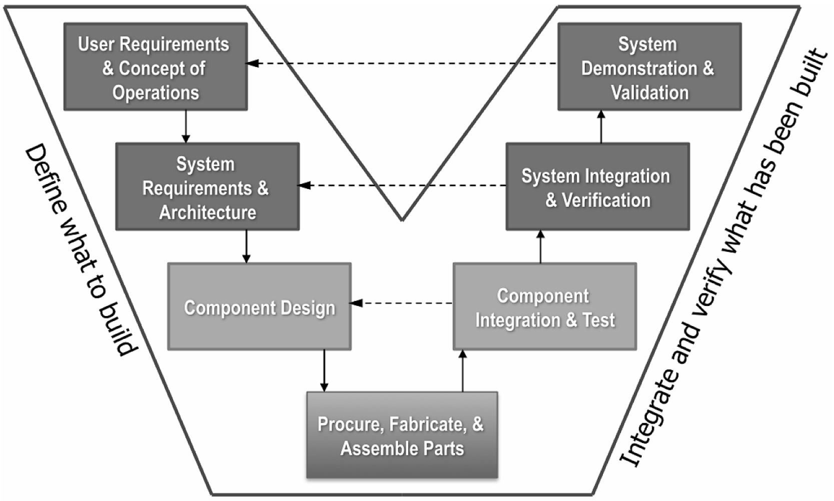

Fig. 1. Typical systems engineering V-model.

traditional approach in engineering has been a reductionism and discovery approach to understand what makes a system function. If we were to take this same approach to our understanding of integration, the question becomes how do we divide and conquer integration? Moreover, what are the tools and practices that are involved in determining the integration maturity of an extremely complex system? In SE and project management a fundamental practice for determining effectiveness, efficiency, and direction is through the use of metrics.

In order to address the concerns relevant to engineering and managing integration, we are proposing an Integration Readiness Level (IRL) metric for a systematic measurement of the interfacing of compatible interactions for various technologies and the consistent comparison of the maturity between integration points. We will present the theory behind the development of this metric and how it compares to other metrics for system integration management. We then use IRL and three other well documented integration metrics to describe the integration failure of three well known aerospace projects. We will use these case analyses to demonstrate how these integration metrics apply different theories that can provide richer insights for the analysis of integration. We then expand upon this work with the objective of presenting a verified and validated IRL and supporting "checklist" based on a survey to assess the criticality of decision criteria in the "checklist." We conclude with a discussion of the implications of our IRL to the practice of systems engineering and aerospace and how this may lead to additional questions for further investigation.

2. Development of an integration maturity metric

2.1. Why integration maturity metrics?

The use of technology maturity metrics within aerospace has been around since the introduction of Technology Readiness Level (TRL) in the 1980's, and is a fairly mature practice. Yet, the emergence of large, complex systems created through the integration of diverse technologies has created the need for a more modern maturity metric [15]. For example, complex system development and integration has too often posed significant cost, schedule and technical performance risks to program managers, systems engineers, and development teams. Many risk factors have played a key role in degrading this process, but acceptable technology maturity has often been the principal driver, particularly in programs where innovation is fundamental to stakeholder requirements. The path of least resistance to this would be to simply use an already existing metric that is able to provide for an effective solution. Initially, TRLs seem to provide this capability. They are ambiguous, yet descriptive; applied at many different levels of system development; and start at concept definition and move all the way through mission/flight proven technology in the intended environment [22]. TRLs were originally developed by the United States (US) National Aeronautics and Space Administration (NASA) to rate the readiness of technology for possible use in space flight [23]. Later, the US Department of Defense (DoD) began using TRL to assess new technology for insertion into a weapon system [13]. With TRL's widespread use within NASA and the DoD, other US government agencies and their contractors (e.g. Department of Energy (DoE), Sandia National Laboratory) have also adopted the TRL scale. Today, TRLs provide critical functionality in the decision making and developmental control of projects at both NASA and DoD [9,22,28]. In fact, in some organizations, different labs, departments, and groups have been organized with responsibility for bringing new technologies through the various TRL levels, handing off to each other as the technology matures [9,23,32]. Additionally, in the years following the introduction of TRL, a variety of other maturity metrics have been proposed as decision support tools for acquisitions (e.g. Design Readiness Level, Manufacturing Readiness Level; Software Readiness Level; Operational Readiness Level; Human Readiness Levels; Habitation Readiness Level; Capability Readiness Levels [3,6,7]).

Smith [37] identified that TRLs are unable to assess maturity at the system level and that they tend to distort many different aspects of readiness into one single number, the most problematic being integration. The solution he proposed involves using orthogonal metrics in combination with a Pair-wise Comparison Matrix in order to compare equivalent technologies for insertion into a system. His approach is specific to the domain of Non-Developmental Item (NDI) software for acquisition normally into defense related systems. While Smith's solution may be sophisticated and mathematically based, it does not specifically address the maturity of integration. He views integration as being binary, either a technology is integrated or it is not, and integration is simply part of what he terms the 'overall environmental fidelity'. It may be the case with NDI software that integration is binary; however, as will be demonstrated by the case studies presented in this paper, integration is not always a binary act and must be matured, just as technology is itself.

Mankins [23] identified TRL's inability to measure the uncertainty involved when a technology is matured and integrated into a larger system. He points out that TRL is simply a tool that provides basic guideposts to component technology maturation, independent of system specific integration concerns. Also, TRL does not denote the degree of difficulty in moving from one TRL to the next for a specific technology. Mankins' solution to this problem was the Integrated Technology Analysis Method (ITAM) which was originally developed by NASA in the mid-1990s. The basic concept behind ITAM is the Integrated Technology Index (ITI) which is formulated from various metrics including delta-TRL, the difference in actual to desired TRL, research and development effort required, and technology criticality. While ITAM and ITI attempt to provide an estimate of difficulty in system development from a technology maturation viewpoint, Mankins points out that the approach is not always appropriate. Graettinger, et al. [14] states that while TRLs measure technology maturity, there are many other factors to consider when making the decision to insert a technology into a system. This implies that while two technologies might have equivalent TRLs, one may more readily integrate into the system environment. In addition, it is observed that TRL's greatest strength is to provide an ontology by which stakeholders can commonly evaluate component technologies. While it is true that in practice TRLs may not be a perfect metric, we must not lose sight that TRL is a tool, and if a tool is used to do something for which it was not created, then there will be errors, setbacks, or even failures. TRL was never meant to evaluate the integration of a given technology with another, and especially not within a large, complex system. Despite TRL's wide use and acceptance there exists very little literature analyzing the effectiveness of TRLs in relation to integration. In addition, the metrics and ontology for the coupling and maturation of multiple technologies and systems has been shown to be an unresolved issue of strategic relevance [26,39]. Finally, component level considerations relating to integration, interoperability, and sustainment become equally or more important from a systems perspective during development [33]. Indeed, Mosher [25] described system integration as the most difficult part of any development program. This limitation in TRL's ability can be filled by another metric specifically geared towards integration readiness assessment.

The application of ontology metrics to support integration has been extensively used in the computer industry to define coupling of components [30,31], but a common ontological approach to technology integration for system development has been far less developed. In order to clarify what an integration maturity metric should provide we conducted a review of the literature that encompassed both work done on integration maturity metrics and the practice and lessons learned about TRLs' use in government and industry. We concluded that an effective integration metric must be considered from both the lowest level (e.g. configuration item) to the system level. We contend that this can only be accomplished through the use of a metric that describes integration in general enough terms to be used at each level, but specific enough to be practical. We concluded that the limitations found in TRL can be translated into requirements for an Integration Maturity Metric (IMM). These limitations include:

- Distorts many aspects of technology readiness into one metric, the most problematic being integration [37];
- Cannot assess uncertainty involved in maturing and integrating a technology into a system [7,35–38];
- Does not consider obsolescence and the ability of a less mature technology to meet system requirements [35,37,38]; and
- Unable to meet need for a common platform for system development and technology insertion evaluation [9,11].

If these basic concepts are translated into IMM requirements we can begin to investigate a solution that satisfies these requirements. The IMM requirements are as follows:

1. IMM shall provide an integration specific metric, to determine the integration maturity between two configuration items, components, and/or subsystems.
2. IMM shall provide a means to reduce the risk involved in maturing and integrating a technology into a system.
3. IMM shall provide the ability to consider the meeting of system requirements in the integration assessment so as to reduce the integration of obsolete technology over less mature technology.
4. IMM shall provide a common platform for both new system development and technology insertion maturity assessment.

2.2. *Finding a solution*

TRL was formally purposed in a 1989 *Acta Astronautica* article by Sadin, et al. and was based upon a well known technology maturation model used by NASA at the time [22,32]. Initially, TRL was determined by the assigning of numbers to the phases of technology development, such that management, engineering, and outside vendors could communicate a common language, mainly for contractual purposes. It has become the benchmark and cornerstone of technology acquisition and project budgeting for many government funded technology development projects, but as for integration maturity, the literature reveals limited research that measures integration on any scale.

Before examining our options we must first differentiate what is meant by the term IMM. There is a large number of metrics that can be used to evaluate integration, however, not integration maturity. In addition to the IMM requirements we are seeking a metric that can be understood by all the relevant stakeholders and evaluates integration maturity. One example is the DoD's Levels of Information Systems Interoperability (LISI) which measures aspects of integration such as Processes, Applications, Infrastructure, and Data (PAID) [8]. While these are all critical concepts to consider during integration, methodologies to assess these areas are fairly mature and can be dealt with by information technology practices.

One of the more refined examples is Mankins' Integrated Technology Index (ITI) which he proposes as a method for uncertainty reduction in the developmental effort of space systems. ITI uses the concepts of delta-TRL (ΔTRL), R&D Degree of Difficulty (R&D^3), and Technology Need Value (TNV) to calculate the system ITI, which can then be used to compare and contrast different technologies for insertion/acquisition, see Equation 1. ITI is essentially an average of the product of delta-TRL, R&D^3, and TNV for all the subsystem technologies within a system. By this method the lower the ITI, the lower the overall risk of technology maturity impacting successful system development, integration, and operational deployment.

$$IT1 = \frac{[\Sigma(\#\ \textit{Subsystem Technologies})(\Delta TRL * R\&D^3 * TNV)]}{\textit{Total Number of Subsystem Technologies}} \tag{1}$$

Ultimately, ITI can be used to make management decisions and provides a mathematical base for system-level technology assessment [23]. If we compare ITI to our IMM requirements, Requirement 1 is not met since ITI measures the difficulty of integrating, not the specific integration maturity between component technologies. Requirement 2 is met by ITI's use of R&D effort and delta-TRL as variables which work to reduce the uncertainty involved in system integration. Requirement 3 is not met since ITI has no variable to consider the integrated system's ability to meet system requirements. Finally, Requirement 4 is met because there is no limiting factor that binds ITI to either new system development or technology insertion, as long as it is used as a relative metric.

Another solution proposed by Fang, et al. [12] developed a "Service Interoperability Assessment Model" which is intended to be used as an autonomous assessment model for determining service interoperability in a distributed computing environment. The key aspect of this model is the identification of five levels of integration: Signature, Protocol, Semantic, Quality, and Context. The assessment model uses a weighted sum that calculates what they term K or the degree of interoperability. K is composed of five factors that are normalized assessments of each level of integration. Each factor can be a normalized combination of the other sub-factors, such as semantics, which uses a concept tree to produce mappings between input and output relationships connecting the integrating services, or a subjective scoring. Benchmarking this model against the IMM requirements we find that Requirement 1 is met as the metric explicitly identifies integration maturity between components/sub-systems. Requirement 2 is met by the quantitative assessment of the identified levels of service interoperability for uncertainty reduction. Requirement 3 is not met since the level that might be able to assess the meeting of system requirements is the context level, which the authors specifically identify as being incomplete in its definition. Requirement 4 is met due to clearly defined mathematical scales, with the exception of the context level, which do not limit this metric to any specific integration activity.

Another integration metric developed by Nilsson, et al. [29] was created to assess system-level integration using a four-level scale with each level having multiple strategies/sub-levels to describe how the

Table 1
Nilsson, et al. Breakdown [29]

1. Integration Technology
1.1 Levels of Integration Technology

- *Low* – Unreliable and time consuming data transfer methodology.
- *Medium* – Reliable and effective byte-stream transfer between systems.
- *High* – The use of defined protocols, remote procedure calls, and mechanisms for automated data conversion.

1.2 Strategies for Integration Technology

- *Manual Data Transfer* – Implies the transfer of data is triggered by the user or another component/system.
- *Automatic Data Transfer* – Implies automated data exchange triggered by either scripts or autonomous processes/systems.
- *Common Database* – Implies a common data medium that is shared by all integrating systems.

2. Integration Architecture
2.1 Levels of Integration Architecture

- *Access to User Interface* – "Black Box" functionality, only the user interface is presented to the integrating component/system.
- *Access to Data* – The integrating component/system can access the data of another component/system.
- *Access to Functionality* – The integrating component/system has the ability to execute internal functionality.

2.2 Strategies for Integration Architecture

- *Controlled Redundancy* – More than one component/system stores the data, however the overall data is controlled centrally.
- *Common Data Storage* – All the data is stored and controlled by a single entity.
- *Distributed Data Storage* – More than one component/system stores the data, and overall data control is also distributed.
- *Activating Other Systems* – The integrating component/system can activate other systems and therefore bring additional data online.
- *Abstract Data Types* – Data is stored and controlled both centrally and distributed, but with the addition of tagging such as units, version numbers, modification dates, etc.

3. Semantic Architecture

- *Separate Sets of Concepts* – Systems built by different vendors, lacking a common data translation language.
- *Common Concepts for Common Parts* – System built by one vendor or group of vendors that set a common language for describing data elements.

4. User Integration

- *Accessibility*
  - ∗ One at a Time – The user can control only one component/system independently from the rest.
  - ∗ Simultaneously – The user can control multiple components/systems without any performance/control related issues.
- *User Interface Style*
  - ∗ Different – Each component/system has a different interface.
  - ∗ Common – All components/systems share a common interface.

level could be achieved. Nilsson et al. [29] offer warnings for combinations of sub-metrics that present risk in the integration. The levels with their associated breakdowns are displayed in Table 1.

When compared to the IMM requirements, Requirement 1 is met in that the model is derived from a standard network model, Transmission Control Protocol (TCP), despite the fact that this framework is really applied at the system level. Requirement 2 is met since the direct focus of this work was to reduce uncertainty through the use of the framework, thus making for a more "Integration Friendly" system. Requirement 3 is not met primarily because this work is directed toward user-system integration, not

Table 2
OSI conceptual levels

Level	Conceptual level
7	*Verified and Validated*
6	*Accept, Translate, and Structure Information*
5	*Control*
4	*Quality and Assurance*
3	*Compatibility*
2	*Interaction*
1	*Interface*

Table 3
Integration readiness levels [35]

IRL	Definition
7	The integration of technologies has been *verified and validated* with sufficient detail to be actionable.
6	The integrating technologies can *accept, translate, and structure information* for its intended application.
5	There is sufficient *control* between technologies necessary to establish, manage, and terminate the integration.
4	There is sufficient detail in the*quality and assurance* of the integration between technologies.
3	There is *compatibility* (i.e. common language) between technologies to orderly and efficiently integrate and interact.
2	There is some level of specificity to characterize the *interaction* (i.e. ability to influence) between technologies through their interface.
1	An *interface* (i.e. physical connection) between technologies has been identified with sufficient detail to allow characterization of the relationship.

the meeting of system requirements such as performance, throughput, etc. Requirement 4 is not met since this model primarily deals with developing new systems to be "Integration Friendly", although the authors discuss the possible application to legacy systems as future work. The highest levels are built from the International Standards Organization's TCP (ISO/TCP) model, and the concept of using an open standard as the base of a metric is appealing, as it is built upon an established protocol.

In fact, there exists a standardized model for inter-technology integration, one which starts at the lowest level of integration and moves all the way through to verification and validation. This model is the International Standards Organization's Open Systems Interconnect (ISO/OSI) model, the TCP model is a specific sub-set of the OSI model. This model is used in computer networking to create commonality and reliability in the integration of diverse network systems. The OSI model is 7-layers, or levels, with each level building upon the previous [18].

OSI appears to be a highly technical standard which is solely intended for network system application, yet, if the layer descriptions are abstracted to conceptual levels, this model can describe integration in very generic terms. In fact, these generic terms are the method by which the OSI model is taught and can be found in most computer networking textbooks. Table 2 represents the conceptual levels defined in a fundamental network systems textbook [2].

Using these conceptual levels we, just as Nilson, et al. [29], used a standard (i.e. OSI) as our foundation for developing an IMM. Our initial IMM, termed Integration Readiness Level (IRL), was initially proposed in a previous paper and is summarized in Table 3 [35].

It might appear as though IRL is a completed metric at this point and it should be applied to some examples as to determine its usefulness. However, it is interesting to note that TRL itself started out as a 7-level metric [32].

IMM Requirement 1 is met, since IRL's main concept is to be used to evaluate the integration of two TRL assessed technologies. Requirement 2 is met as IRL 1 through 6 are technical gates that reduce uncertainty in system integration maturity. Requirement 3 is met specifically by IRL 7, which requires the integration to be successful from a requirements standpoint. Requirement 4 is met, but with some uncertainty, since there is no reason that this metric could not be applied to both system development and technology insertion, but it is difficult to know when development should end and operations begin. This is rooted in the fact that the OSI model came from the information technology industry where technology insertion is done almost on a daily basis, just not always successfully. What is truly causing this uncertainty is the fact that IRL has no indication of when the integration is complete. Thus, we have uncovered a problem with our initial IRL, it does not have any measures equivalent to TRLs 7–9 (i.e. environmental testing, operational support, and mission proven maturity). As with the original 7-level TRL scale, IRL only evaluates to the end of development and does not address the operational aspects of the integration. NASA's transition to a 9-level TRL scale was prompted to allow TRL to be used well past the developmental stage. Having identified this deficiency, it is now certain that we do not meet Requirement 4, so IRL must be expanded at least one level to meet Requirement 4.

In expanding IRL, it should be kept in mind that by IRL 7 all of the technical integration issues are resolved, so what we are looking for are qualities that describe operational support and proven integration, into the system environment. Being that IRL was created to be used with TRL and the fact that they share many qualities it makes sense to look at TRLs 8 and 9. TRL 8 is defined as:

> "*Actual system completed and "flight qualified" through test and demonstration* (*ground or space*)*"* [23]

TRL 8 implies at the very least demonstration and testing in the system environment. IRL 8 can be extracted directly from this definition with the only change being in the term "flight". While the original TRLs implied a technology being "ready to fly," it might be the case that the integration is between two pieces of technology, such as ground software, that will never actually "fly" and are simply brought together to fill a need. In this scenario, we are integrating two technologies for a specific mission requirement, so we will change "flight" to "mission" to broaden the possible applications of IRL. Thus IRL 8 is:

> *IRL 8 – "Actual integration completed and "Mission Qualified" through test and demonstration, in the system environment."*

At this point IRL is still not complete, specifically, how do we know when we are fully mature? TRL provides this in TRL 9, and consequently IRL should have the same highest-level maturity assessment. TRL 9 is defined as:

> *"Actual system "flight proven" through successful mission operations*" [23]

Again we will change "Flight Proven" to "Mission Proven" for IRL 9.

> *IRL 9 – "Actual integration "Mission Proven" through successful mission operations"*

As was previously stated, our initial IRL met Requirements 1, 2 and 3. The addition of IRLs 8 and 9 facilitate the satisfaction of Requirement 4. We now have a metric that can completely describe the maturity assessment of the integration of two pieces of technology. Also, since IRL is capable of describing mature system integration well past the developmental stage, it is possible to use it to assess technology insertion into a mature system. Our proposed IRL scale is represented in Table 4. Additionally, IRL is compared and contrasted to the other IMMs along with the IMM requirements in Table 5.

For further clarification, the nine levels of IRL presented in Table 4 can be understood as having three stages of integration definition: semantic, syntactic, and pragmatic. Semantics is about relating meaning

Table 4

Integration readiness levels

	IRL	Definition	Description	Risk of Not Attaining
Pragmatic	**9**	Integration is **Mission Proven** through successful mission operations.	IRL 9 represents the integrated technologies being used in the system environment successfully. In order for a technology to move to TRL 9 it must first be integrated into the system, and then proven in the relevant environment, so attempting to move to IRL 9 also implies maturing the component technology to TRL 9.	The development stage was never completed; this is more of a programmatic risk. However, if the IRL model was used up to this point there should be no (technical) setbacks to stop the integrated system from moving to operational use.
	8	Actual integration completed and **Mission Qualified** through test and demonstration, in the system environment.	IRL 8 represents not only the integration meeting requirements, but also a system-level demonstration in the relevant environment. This will reveal any unknown bugs/defect that could not be discovered until the interaction of the two integrating technologies was observed in the system environment.	The system is still only “Laboratory Proven” and has yet to see real-world use. This can allow unforeseen integration issues to go unattended.
Syntactic	**7**	The integration of technologies has been **Verified and Validated** and an acquisition/insertion decision can be made.	IRL 7 represents a significant step beyond IRL 6; the integration has to work from a technical perspective, but also from a requirements perspective. IRL 7 represents the integration meeting requirements such as performance, throughput, and reliability.	If the integration does not meet requirements then it is possible that something must be changed at a lower level, which may result in the IRL actually going down, however, in most cases the work done to achieve higher levels can be re-implemented.
	6	The integrating technologies can **Accept, Translate, and Structure Information** for its intended application.	IRL 6 is the highest technical level to be achieved, it includes the ability to not only control integration, but specify what information to exchange, unit labels to specify what the information is, and the ability to translate from a foreign data structure to a local one.	The risk of not providing this level of integration could be a misunderstanding of translated data.
	5	There is sufficient **Control** between technologies necessary to establish, manage, and terminate the integration.	IRL 5 simply denotes the ability of one or more of the integrating technologies to control the integration itself; this includes establishing, maintaining, and terminating.	The risk of not having integration control, even in the case of technologies that only integrate with each other, is that one technology can dominate the integration or worse, neither technology can establish integration with the other.
	4	There is sufficient detail in the **Quality and Assurance** of the integration between technologies.	Many technology integration failures never progress past IRL 3, due to the assumption that if two technologies can exchange information successfully, then they are fully integrated. IRL 4 goes beyond simple data exchange and requires that the data sent is the data received and there exists a mechanism for checking it.	Vulnerability to interference, and security concerns that the data could be changed if part of its path is along an unsecured medium

Table 4, continued

	IRL	Definition	Description	Risk of Not Attaining
Semantic	**3**	There is **Compatibility** (i.e. common language) between technologies to orderly and efficiently integrate and interact.	IRL 3 represents the minimum required level to provide successful integration. This means that the two technologies are able to not only influence each other, but also communicate interpretable data. IRL 3 represents the first tangible step in the maturity process.	If two integrating technologies do not use the same data constructs, then they cannot exchange information at all.
	2	There is some level of specificity to characterize the **Interaction** (i.e. ability to influence) between technologies through their interface.	Once a medium has been defined, a “signaling” method must be selected such that two integrating technologies are able to influence each other over that medium. Since IRL 2 represents the ability of two technologies to influence each other over a given medium, this represents integration proof-of-concept.	The risks of not attaining, or attempting to skip, this level can be data collisions, poor bandwidth utilization, and reduced reliability of the integration
	1	An **Interface** between technologies has been identified with sufficient detail to allow characterization of the relationship.	This is the lowest level of integration readiness and describes the selection of a medium for integration.	It is impossible to have integration without defining a medium, so there are no real risks here, however, the selection of a poor medium may impact performance requirements later on.

Table 5

IMM summarization

Metric	Concept	IMM Req. 1	IMM Req. 2	IMM Req. 3	IMM Req. 4	Strengths	Weaknesses
ITI	"quantitative measure of the relative technological challenge inherent in various candidate/competing advanced systems concepts" [23]	–	+	–	+	– Quantitative use of TRL in addition to two other proven and common metrics. – Has been successfully implemented in the NASA Highly Re-useable Space Transportation study [23] – Able to provide a relative measure of what technology to insert/integrate.	– Does not consider Cost/Schedule parameters – Requires Work Breakdown/System Architecture to be complete and accurate prior to assessment. – Requires TRL assessment prior to ITI assessment.
Fang et. al.	"a reference assessment model of service interoperability by fuzzy quantization of the interoperability." [12]	+	+	–	+	– Quantitative assessment process. – Able to provide a relative measure of what technology/service provides the best interoperability. – Transforms technical data into metric for managerial decisions.	– Rigorous mathematical assessment requires much data on the integration to be gathered prior to assessment. – Does not consider technology readiness/maturity. – Metric was designed for *service* integration, not *system* integration.
Nilsson et. al.	System integration can be improved if it is split into four "aspects of integration": Integration technology, Integration architecture, Semantic Integration, and User Integration [29]	+	+	–	–	– Based on an open, widely accepted standard (ISO/TCP). – Provides strategies for improving integration. – Does not require concise data on the integration.	– Purely subjective assessment process. – Does not consider technology readiness/maturity. – Not really created as a metric, simply an attempt at organizing system integration.
IRL	IRL is a metric that is to be used to evaluate the integration readiness of any two TRL-assessed technologies.	+	+	+	+	– Based on an open, widely accepted standard (ISO/OSI). – Technology readiness is incorporated in the overall assessment. – Subjective assessment made on technical data.	– Requires Work Breakdown/System Architecture to be complete and accurate prior to assessment. – Requires TRL assessment prior to IRL assessment.

Fig. 2. Application of IRL.

with respects to clarity and differentiation. Thus IRL 1–3 are considered fundamental to describing what we define as the three principles of integration: interface, interaction, and compatibility. We contend that these three principles are what define the subsistence of an integration effort. The next stage is Syntactic, which is defined as a conformance to rules. Thus IRLs 4–7 are about assurance that an integration effort is in compliance with specifications. The final stage is Pragmatic, which relates to practical considerations. Thus, IRLs 8–9 are about the assertion of the application of an integration effort.

Figure 2 represents the application of IRL. It is to be used to assess integration maturity between two TRL assessed technologies. The combination of TRL and IRL creates a fast, iterative process for system-level maturity assessment. IRL must now be applied to some real-world case studies, and the results interpreted.

In the following sections we will present three aerospace cases that had documented integration issues and analyze how the four integration metrics can describe the integration. We selected these cases because they encompass multiple types of systems and multiple vendors.

3. Case studies using IMM

3.1. Mars climate orbiter

Mars Climate Orbiter (MCO) crashed into the Martian atmosphere on September 23, 1999. The Mishap Investigation Board (MIB) found the reason for the loss of the spacecraft to be the use of English units in a ground software file called "Small Forces", the output of which was fed into another file called Angular Momentum Desaturation (AMD) which assumed inputs to be in metric, as were the requirements of the project. The MCO navigation team then used the output of this file to derive data used in modeling the spacecraft's trajectory, and perform corrective thruster firings based upon these models. The Software Interface Specification (SIS) defined the format of the AMD file and specified that the data used to describe thruster performance, or the impulse-bit, be in Newton-Seconds (N-s) [19]. While this was the case with the AMD file aboard the spacecraft, which gathered data directly from onboard sensors, the ground AMD file was populated with data from the "Small Forces" file, which outputs its measurements in Pound-seconds (lbf-s). Since 1 lbf-s is equal to 4.45 N-s, there existed a 4.45 error-factor in what the navigation team observed and what the spacecraft actually did. When the navigation team observed the lower numbers being produced by their software, they assumed the spacecraft was going off course and immediately began firing the thruster with 4.45 times more force than was necessary. The thruster firing occurred many times during the 9-month journey from Earth to Mars and resulted in MCO entering Martian Orbital Insertion (MOI) at 57 kilometers altitude, rather than the objective of 226 kilometers. The minimum survivable altitude was calculated to be 80 kilometers, below which the effects of Martian-Gravity would cause the orbiter to burn up in the atmosphere.

The MIB found eight contributing causes that either contributed to, caused, or failed to catch the error in the ground software. While many of these were identified as organizational issues associated with NASA and the contractors working on MCO, two are of interest to this research effort. The first is listed as "System engineering process did not adequately address transition from development to operations" [27,

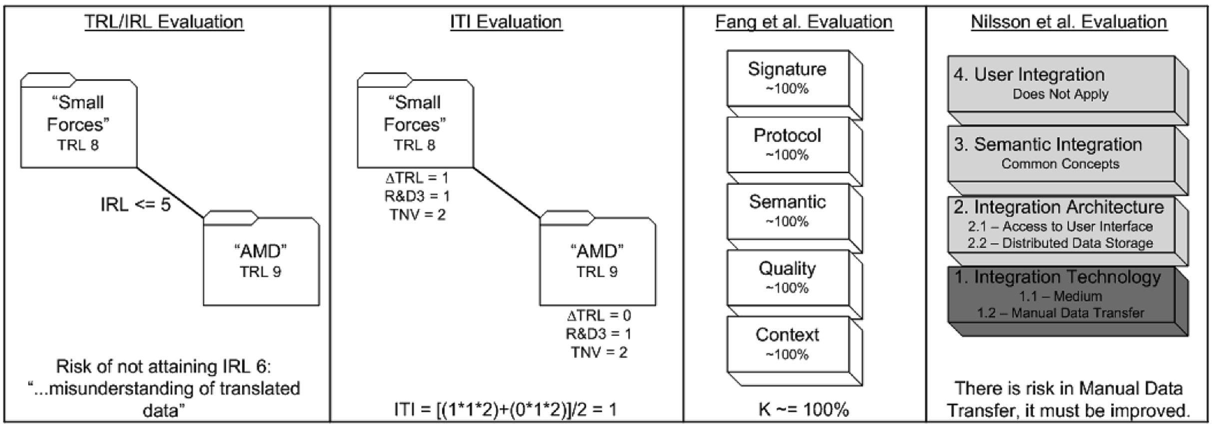

Fig. 3. MCO Metric Analysis.

pg 7]. This suggests that the MIB was unsatisfied in how the SE process determines system operational maturity. The second is similar, but more directed toward integration, "Verification and Validation process did not adequately address ground software" [27, pg 7]. This suggests that some sort of testing process should have been in place to catch the units error. While testing is absolutely necessary, it is not always capable of catching the many small errors that can occur when two different pieces of software and/or hardware exchange data in a raw format. If the integration of two pieces of technology followed some sort of maturation process, just as the technology itself does, this would provide an assessment of integration readiness and a direction for improving integration during the development process.

In our assessment of MCO we evaluated the integration of the ground software "Small Forces" and "AMD" files as they were the primary technical failure leading to MCO's demise. Figure 3 is a summary of how the four IMMs could evaluate MCO.

IRL has uncovered the basic problem in MCO, a misunderstanding of translated data, or in MCO's case un-translated data. None of the other metrics catch major risks or issues with the maturity of MCO's ground data files. Nilsson et al. only catches the risk of manual data transfer present in the emailing of data files between project teams.

3.2. ARIANE 5

The ARIANE series of launch vehicles was developed by the European Space Agency (ESA) as a commercially available, low cost, and partially reusable solution for delivering payloads into Earth orbit. ARIANE 5's maiden flight occurred on June 4, 1996 and ended 37 seconds after liftoff, when the vehicle veered off of its flight path and disintegrated due to aerodynamic stress. In the following days an independent inquiry board was established to determine the cause of the failure [21].

The board found that the failure began when the backup Inertial Reference System (SRI) failed 36.7 seconds after H0 (H0 is the point in time at which the main engines were ignited for liftoff) due to a software exception. Approximately 0.05 seconds later the active SRI went offline due to the same software exception, and began dumping diagnostic information onto the databus. The On-Board Computer (OBC) was reading the databus and assumed the data was the correct attitude data, and since both SRIs were in a state of failure, it had no way to control data on the bus. The misinterpreted diagnostic data was factored into calculations for thruster nozzle angles by the OBC. The incorrect thruster nozzle angles forced the launcher into an angle-of-attack that exceeded 20 degrees, leading to aerodynamic stress that

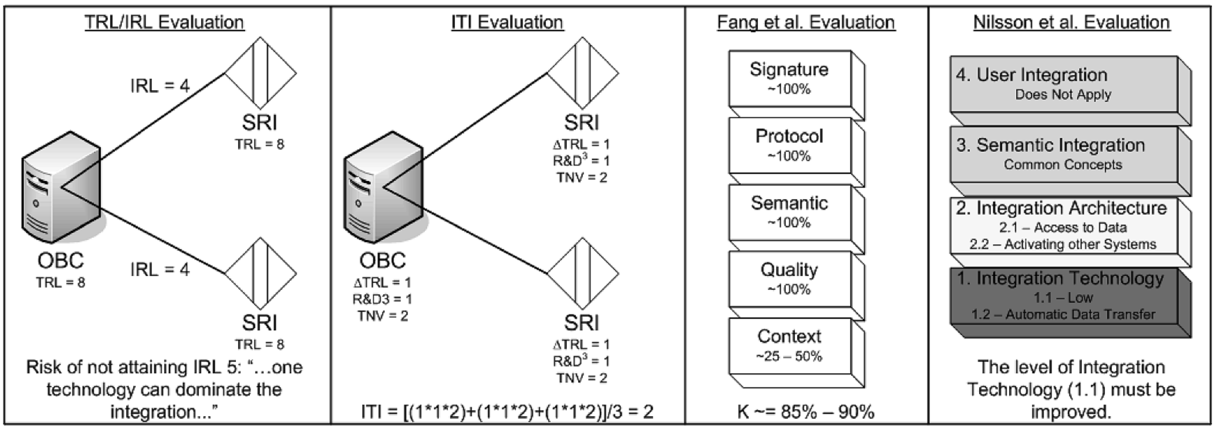

Fig. 4. ARIANE 5 Metric Analysis.

resulted in the booster engines prematurely separating from the main stage, which finally triggered the self-destruct of the main stage at approximately H0 + 39s [21].

After recovering both SRIs from the crash site, data recovery efforts found that the software exception was caused during the conversion from a large 64-bit floating point number to a 16-bit signed integer. The variable that was converted was the horizontal velocity of ARIANE 5, which was within thresholds for the proposed trajectory. The reason the software was unable to handle the high velocity value was due to the fact that it was unchanged from ARIANE 4, whose horizontal velocity never approached the values ARIANE 5 was built to achieve. If either of the SRIs had ignored the software exception, an ability that they had, the launcher would have continued functioning flawlessly [21].

For ARIANE 5 we examined the integration between the two SRIs and the OBC, we will assume that the software exception is unpreventable and thus examine how the integration maturity affected the OBC's ability to function. Figure 4 is a summary of how the four IMMs could evaluate ARIANE 5.

Based on these evaluations a few conclusions can be drawn. First, IRL recommends that there should be some form of integration control, otherwise one technology could dominate the integration, and this is exactly what occurred with ARIANE 5. The OBC had no way to control what data the SRIs dumped onto the databus. Nilsson et al. highlights risk in a low level of Integration Technology coupled with Automated Data Transfer, this metric suggests that for Automated Data Transfer a protocol, i.e. high, level of Integration Technology should be used to avoid errors or misinterpreted data. One caveat is that a protocol adds overhead, which influences performance, and this metric does not evaluate that. ITI indicates low risk in the integration of the OBC with the SRIs. Since ITI is a relative measure, we will create a 'new' SRI that does not suffer from the same software exception, therefore, the highest TRL it could achieve is 7, so ΔTRL $= 2$ (we want to reach TRL 9), R&D^3 $= 2$ (the new SRI requires new development, but it is based on the old SRI), TNV $= 1$ (the old SRI is now an option). Calculating ITI now yields 3.33. What is revealed is that ITI indicates less risk in the old SRI, which is logical since ITI calculates integration maturity based on technology maturity. Fang et al. [12] indicates some risk at the context level, simply due to the fact that the SRIs are basically sensors, and the metric specifically speaks of sensors being a context level concern, even though the context level is unfinished.

3.3. Hubble space telescope – Part 1

From the identification of the need for a space-based observatory in 1923 to the latest on-orbit retrofitting in March 2002 by STS-109, the SE behind Hubble has been nearly flawless, even in the face

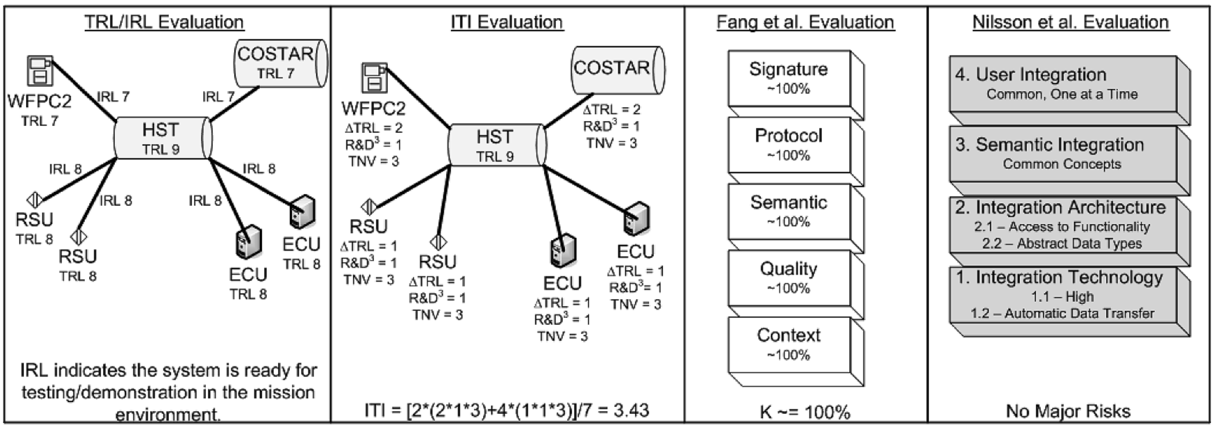

Fig. 5. Service Mission 1 Metric Analysis.

of daunting challenges and unforeseeable obstacles [24]. In this brief case example we will examine how much of the technical success the Hubble program has enjoyed is a direct result of management of the integration.

From its beginning HST was envisioned as an upgradeable space observing platform, meaning that in addition to being the most detailed optical telescope in history, it was also built to carry additional science equipment, and was built such that this equipment could be added, replaced, or maintained by an extra-vehicular activity event. In order to meet this requirement HST was designed to be as modular as possible, and the outer-shell even included such details as handholds for servicing astronauts. This modularity and upgradeability would soon prove their value as, after initial launch and the start of science operations, it was discovered that the primary mirror had an aberration that caused a blurring of far-off objects, the very objects that were HST's primary objective to study. It seemed as though HST was a failure, however, in December 1993 the STS-61 crew was able to successfully attach Corrective Optics Space Telescope Axial Replacement (COSTAR) to correct the problem, in addition to performing other minor repairs such as fully unfolding a solar panel, tightening bolts, and securing access doors. That mission was designated SM-1 (Servicing Mission – 1), SM-2 occurred five years later and it significantly upgraded the scientific capabilities of HST, in addition to providing minor maintenance support. SM-3A and SM-3B occurred in December 1999 and March 2002 respectively and also enhanced scientific capability while also extending HST's life expectancy. NASA seemed to have found a successful recipe for HST success and longevity. In fact, HST has been the most successful project in NASA history from the perspective of the amount of scientific knowledge gained [20,24].

The primary mirror aberration was undiscovered due to poor testing at the manufacturer, which did not detect the imperfection despite it being outside of acceptable tolerances. The reason it was not detected during the original integration of the optical system was that it was small enough to not be detected in ground testing, and testing to the extent that would have detected it required a space environment where atmospheric distortion was not a factor [24].

Hubble is an incredibly complex architecture that has changed and evolved over time, for simplicity we will examine the integration of SM-1 components, since this mission represents a significant contribution to HST's success. Figure 5 represents the SM-1 assessment with the four IMMs.

HST SM-1 demonstrates that IRL is able to identify a successful architecture. The integrations of the COSTAR and WFPC2 need to be matured further, but they must be integrated into the system and used in the mission environment in order to accomplish this, which is exactly what was done. ITI also indicates low risk, however it is interesting to note that ITISM-1 > ITIARIANE5 > ITIMCO. This seems to indicate more risk in the HST SM-1 architecture, which was a success, as opposed to MCO and ARIANE 5, which were failures. Of course, HST SM-1 is a much more complex evaluation as compared to MCO and ARIANE. The other two metrics basically indicate no real risk, however, there is not enough information available to accurately assess Fang et al. at all levels.

3.4. Hubble space telescope – Part 2

As of 2007, HST had surpassed its expected lifetime and was slowly dying in space, its gyroscopes were approaching the end of their lifecycle, its batteries were nearing their lower levels of acceptable performance, and the fine-guidance sensors had already begun failing. If HST was not serviced and the batteries ran out, or navigational and attitude control was lost, certain instruments aboard could be permanently damaged either due to low temperatures or direct exposure to sunlight. Meanwhile, demand for scientific time with the telescope had only increased since its inception, while the data rate with which HST delivers new information had increased by a factor of 60 due to upgrades during SM-3B. NASA has since performed SM-4 to keep Hubble operating well into the future, at a great risk to human life due to Hubble's high orbit. The Columbia Accident Investigation Board (CAIB) requirements for shuttle operations highlighted the risk of HST SM-4 since time and fuel considerations would not allow the crew to safely dock to the International Space Station (ISS) if damage was found on the heat shield. To combat this problem a robotic servicing mission (RSM) had been suggested; an independent committee was established to determine the feasibility of this mission, their findings include [20]:

- The TRLs of the component hardware/software are not high enough to warrant integration into a critical mission system, these include the LIDAR for near-field navigation at TRL 6, due to not being proven in a space-environment.
- The more mature components such as the Robotic Manipulator System (RMS) and the Special Purpose Dexterous Manipulator (SPDM) are space-proven but have never been integrated in a space-environment.
- The ability of a robotic system to capture and autonomously dock to an orbiting spacecraft has never been tested under the conditions which HST presents.
- The technology and methodology of shuttle docking and service of HST has been proven and the risks to the crew and vehicle are no greater than any other mission.
- While a shuttle mission to HST would be costly and interrupt the already delayed construction of the ISS, HST's value and contributions to the scientific community greatly outweigh these.

Due to the timeframe of HSTs impending failure, which was calculated to occur sometime between 2008 and 2010, a robotic mission was not be feasible [20]. What is interesting is that an independent committee has considered more than simply technology readiness in their assessment of the options. In fact, they specifically speak of the integrated performance of all the components involved. Furthermore, some of the TRLs of the components will be matured in other space bound systems, such as the United States Air Force's XSS-11, which will move the TRL of the LIDAR past TRL 6, but its specific integration maturity will be unchanged [20].

The previous case studies have been conducted on operational systems; in this examination we will assess the integration of the key technologies involved in the hypothetical RSM development. This will highlight the technologies and integrations that must be matured for RSMs to be possible in the future. Figure 6 represents the approximate, simplified system architectural analysis of the dexterous robot envisioned to service HST [20].

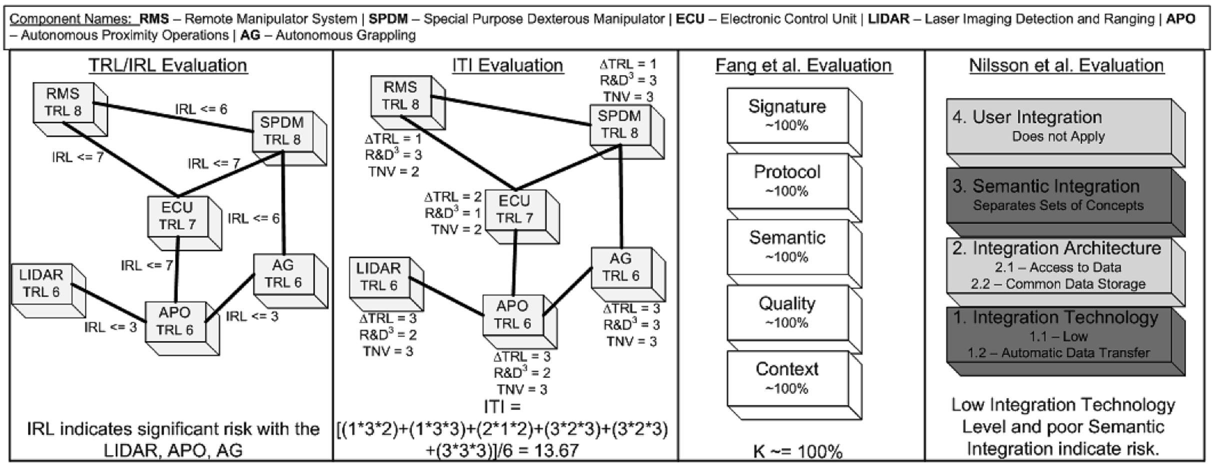

Fig. 6. HST Dexterous Servicing Robot Architecture Metric Analysis.

The evaluation of the HST RSM provides some interesting insights. First, not only do both ITI and IRL indicate risk in the maturity, but both highlight the same components/integrations as being in need of further maturity. Nilsson et al. also highlights the fact that the use of different vendors on this project has caused separate sets of concepts to be used, the solution here would be a standards document that could be shared between all stakeholders. Also, risk is present in the low level of Integration Technology. Once again, there is not enough data present to fully evaluate the Fang et al. metric.

3.5. Case study summary

The case studies have provided some insight into how each of the metrics assesses integration maturity, risk, and operational readiness. Table 6 is a summary of what the case studies revealed about each metric.

4. Development of a guide for IRL

In the previous section we have described the development of an IRL based on a set of IMM requirements. In this section we will give further explanation of this IRL as we begin the development of a verified and validated set of IRL metrics that can be useful in developing a more comprehensive systems maturity assessment methodology that addresses the complexity of integration in a less heuristic or subjective manner. In the context of this effort, verification addresses whether or not the correct IRLs were identified/defined and validation addresses the relevance or criticality of each IRL. Thus, in creation of the IRL checklist, we used two forms of assessment to specify the decision criteria that may define each IRL: (1) review of systems engineering and acquisition standards, policy, research, and other guidance documents (e.g. DoD 5000.02, INCOSE Systems Engineering Handbook, IEEE 15288, NASA Systems Engineering Handbook), and (2) discussions with subject matter experts (SME) in systems engineering, program management, and acquisition across government, industry, and academia. In all cases an effort was made to capture those documents (e.g., Systems Engineering Plan, TEMP) or document content (e.g., requirements, architecture, compatibility, interoperability, etc.) deemed most significant to an assessment of integration maturity. What resulted was a list of decision criteria for each IRL as shown in Tables 8–16. It should be emphasized that the list of maturity metrics under each IRL is not in order

Table 6

Case Study Summary

Case	ITI	Fang et al.	Nilsson et al.	IRL
MCO	ITI indicates very little risk in the MCO ground software. ITI = 1.	Value of K is approximately 100%. No risk.	Identified issues with manual data transfer, and the lack of units attached to impulse bit, which were the exact technical causes of MCO's failure.	Low IRL (IRL 5) indicates risk in the exchange of impulse bit data. Risk of not attaining IRL 6: "…misunderstanding of translated data"; the cause of MCOs failure.
ARIANE 5	ITI indicates that the old SRIs from ARIANE 4 will integrate better than a new SRI built to ARIANE 5 requirements. ITI = 2.	Some risk indicated, but requires more data for accurate assessment.	Metric suggests the need for a higher level of technical integration between components, but does not consider performance affects of those protocols.	IRL 4 would have indicated and identified risk. IRL 5: Control, risk of not attaining: "…one technology can dominate the integration…"
HST 1	ITI indicates low risk; however an ITI of 3.43 is higher than the ITI for MCO and ARIANE 5, despite HST-SM1 being a successful integration.	Value of K is approximately 100%. No risk.	This metric indicated no real risk.	IRL displayed a relatively mature architecture. Since HST-SM1 was a success this is to be expected.
HST 2	ITI indicates significant risk; this is where ITI's strengths are apparent due to the large R&D effort required for successful technology/integration maturation. ITI = 13.67	Value of K is approximately 100%. No risk.	Uncovered low technical integration and the use of multiple vendors as a significant risk.	IRL suggests significant risks involved in integration maturity. It highlights the very same technologies ITI does as being the most troublesome; this is due to TRL being the basis of both metrics.
Summary	ITI is limited in indicating any risk or need for maturity when the component technologies are themselves mature. ITI does however uncover the difficulty involved in maturing a technology, and the risks if it is the only option.	This metric provides in-depth, quantitative and descriptive analysis; however, the amount of data required to accurately assess a system introduces more complexity and work than is necessary. Also, the final assessment may not be interpreted equally among all stakeholders, and does not provide direction for further maturation.	This metric successfully identified most of the risks involved in the integration maturity; with the only exception being the ability of the integration to meet system requirements. However, this work was not originally intended to be used as a metric, only straightforward ontology.	IRL is able to successfully identify the risks involved in the integration maturity both at the component and system levels. IRL also provides a common language for all stakeholders. What IRL lacks is the ability to measure the difficulty in maturing a technology or integration, such as cost, R&D effort, and/or criticality. Also, IRL requires a system-level quantitative assessment, for complex net-centric systems.

Table 7
Demographics of subject matter experts

Sector	Sample	Years of experience				
		0–5	5–10	10–15	15–20	20+
Government	13	2	2	1	1	7
Industry	20	3	9	2	2	4
TOTAL	**33**	**5**	**11**	**3**	**3**	**11**

Table 8
IRL 1 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 1 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
1.1 Principal integration technologies have been identified	0.58	0.33	0.03	0.06	0.00	0.91	0.09
1.2 Top-level functional architecture and interface points have been defined	0.39	0.52	0.06	0.03	0.00	0.91	0.09
1.3 Availability of principal integration technologies is known and documented	0.15	0.39	0.36	0.06	0.03	0.55	0.42
1.4 Integration concept/plan has been defined/drafted	0.18	0.45	0.21	0.12	0.03	0.64	0.33
1.5 Integration test concept/plan has been defined/drafted	0.12	0.36	0.33	0.18	0.00	0.48	0.52
1.6 High-level Concept of Operations and principal use cases have been defined/drafted	0.06	0.21	0.55	0.15	0.03	0.27	0.70
1.7 Integration sequence approach/ schedule has been defined/drafted	0.06	0.36	0.33	0.21	0.03	0.42	0.55
1.8 Interface control plan has been defined/drafted	0.03	0.12	0.67	0.18	0.00	0.15	0.85
1.9 Principal integration and test resource requirements (facilities, hardware, software, surrogates, etc.) have been defined/identified	0.09	0.36	0.30	0.18	0.06	0.45	0.48
1.10 Integration & Test Team roles and responsibilities have been defined	0.12	0.24	0.33	0.24	0.06	0.36	0.58

of criticality. It should also be emphasized that the lists are not considered to be comprehensive or complete; they are merely an attempt to capture some of the more important decision criteria associated with integration maturity in order to afford practitioners the opportunity to assess the criticality of each decision criteria relative to the IRL it is listed under.

Thus, to establish further verification and validation to the decision criteria, we deployed a survey that asked Subject Matter Experts (SMEs) to evaluate each decision criteria in the context of its criticality to the specified IRL. The criticality criteria for assessing the IRL decision criteria were defined as:

- Critical – IRL cannot be assessed without it
- Essential – without it, IRL can be assessed but with low to medium confidence in the results
- Enhancing – without it, IRL can be assessed with medium to high confidence in the results
- Desirable – without it, IRL can be assessed with very high confidence in the results
- N/A – the metric is not applicable to the IRL assessment

Table 9
IRL 2 Decision criteria and criticality assessment

IRL 2 decision criteria	Relative frequency (RF); $n = 33$					Cumulative RF	
	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
2.1 Principal integration technologies function as stand-alone units	0.18	0.27	0.24	0.30	0.00	0.45	0.55
2.2 Inputs/outputs for principal integration technologies are known, characterized and documented	0.52	0.36	0.06	0.06	0.00	0.88	0.12
2.3 Principal interface requirements for integration technologies have been defined/drafted	0.39	0.33	0.24	0.03	0.00	0.73	0.27
2.4 Principal interface requirements specifications for integration technologies have been defined/drafted	0.27	0.45	0.24	0.03	0.00	0.73	0.27
2.5 Principal interface risks for integration technologies have been defined/drafted	0.06	0.24	0.61	0.09	0.00	0.30	0.70
2.6 Integration concept/plan has been updated	0.06	0.42	0.42	0.09	0.00	0.48	0.52
2.7 Integration test concept/plan has been updated	0.09	0.27	0.52	0.12	0.00	0.36	0.64
2.8 High-level Concept of Operations and principal use cases have been updated	0.12	0.18	0.45	0.21	0.03	0.30	0.67
2.9 Integration sequence approach/schedule has been updated	0.09	0.27	0.45	0.18	0.00	0.36	0.64
2.10 Interface control plan has been updated	0.06	0.30	0.61	0.03	0.00	0.36	0.64
2.11 Integration and test resource requirements (facilities, hardware, software, surrogates, etc.) have been updated	0.15	0.39	0.27	0.15	0.03	0.55	0.42
2.12 Long lead planning/coordination of integration and test resources have been initiated	0.12	0.30	0.30	0.24	0.03	0.42	0.55
2.13 Integration & Test Team roles and responsibilities have been updated	0.03	0.15	0.58	0.21	0.03	0.18	0.79
2.14 Formal integration studies have been initiated	0.12	0.33	0.21	0.21	0.12	0.45	0.42

We sampled 33 SMEs from government and industry with experience in systems engineering, software engineering, program management, and/or acquisition. Table 7 indicates the demographics of the 33 SMEs with respects to years of experience and employment in government or industry. Of these, 85% had greater than five years experience and 33% had greater than 20 years of experience.

For each decision criteria we calculated the relative and cumulative frequencies of the criticalities (reported in Tables 8–16). Relative frequency is the proportion of all responses in the data set that fall in the category (i.e. decision criteria for any IRL). Cumulative relative frequency allows for additional information to be understood about the sensitivity of the response frequency based on a class interval (i.e. Critical/Essential versus Enhancing/Desirable). This is meant to help to identify whether the criticality categories originally identified are too fine and should be modified.

Table 10
IRL 3 decision criteria and criticality assessment

IRL 3 decision criteria	Relative frequency (RF); $n = 33$					Cumulative RF	
	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
3.1 Preliminary Modeling & Simulation and/or analytical studies have been conducted to identify risks & assess compatibility of integration technologies	0.18	0.36	0.45	0.00	0.00	0.55	0.45
3.2 Compatibility risks and associated mitigation strategies for integration technologies have been defined (initial draft)	0.09	0.39	0.52	0.00	0.00	0.48	0.52
3.3 Integration test requirements have been defined (initial draft)	0.15	0.48	0.24	0.12	0.00	0.64	0.36
3.4 High-level system interface diagrams have been completed	0.48	0.27	0.24	0.00	0.00	0.76	0.24
3.5 Interface requirements are defined at the concept level	0.24	0.70	0.06	0.00	0.00	0.94	0.06
3.6 Inventory of external interfaces is completed	0.24	0.33	0.42	0.00	0.00	0.58	0.42
3.7 Data engineering units are identified and documented	0.06	0.45	0.24	0.21	0.03	0.52	0.45
3.8 Integration concept and other planning documents have been modified/updated based on preliminary analyses	0.18	0.27	0.42	0.09	0.03	0.45	0.52

4.1. Semantic (IRL 1-3)

This is the stage at which we fundamentally define the integration needs and the manner in which it will take place. From Tables 8–10 we observe that in IRLs 1–3 a single decision criterion for each IRL is rated as critical by the respondents. For IRL 1 this is *1.1 Principal integration technologies have been identified*. This can indicate that at this level of maturity the criticality of the integration is in the proper identification of the technologies to be integrated.

Obviously, identifying integration elements is the first step in successful integration. Though it may seem trivial, this activity is indispensable as unknown or undefined elements can derail a project that is well along in the development process. Application of proper time and resources at this stage is essential in order to build a proper foundation for future planning and maturation activities. For IRL 2, we observe that the criticality has transferred to an understanding of the input/output (I/O) for the integration.

With the elements of the system integration effort defined at IRL 1 the next step logically moves on to the definition of the I/O requirements of the system. This was identified by SMEs as a critical step and is needed in order to understand the type and complexity of the integrations between technology elements. Indeed, all integration is not the same and survey results show that successful system integration is highly dependent on the accurate understanding of the degree of work needed to successfully connect disparate systems. This information then drives factors such as the application of cost, schedule, and resources during later development activities.

At IRL 3, the data denotes an importance in the diagramming of the system interfaces. To reach this stage of maturity requires leveraging all of the information defined previously. The identified technologies can be mapped and the I/O requirements are drivers for how those elements are to be connected. At this

Table 11
IRL 4 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 4 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
4.1 Quality Assurance plan has been completed and implemented	0.18	0.27	0.36	0.15	0.03	0.45	0.52
4.2 Cross technology risks have been fully identified/characterized	0.12	0.52	0.33	0.03	0.00	0.64	0.36
4.3 Modeling & Simulation has been used to simulate some interfaces between components	0.06	0.24	0.70	0.00	0.00	0.30	0.70
4.4 Formal system architecture development is beginning to mature	0.09	0.52	0.36	0.03	0.00	0.61	0.39
4.5 Overall system requirements for end users' application are known/ baselined	0.24	0.55	0.15	0.06	0.00	0.79	0.21
4.6 Systems Integration Laboratory/Software test-bed tests using available integration technologies have been completed with favorable outcomes	0.09	0.52	0.36	0.03	0.00	0.61	0.39
4.7 Low fidelity technology "system" integration and engineering has been completed and tested in a lab environment	0.06	0.36	0.52	0.06	0.00	0.42	0.58
4.8 Concept of Operations, use cases and Integration requirements are completely defined	0.12	0.30	0.55	0.00	0.03	0.42	0.55
4.9 Analysis of internal interface requirements is completed	0.09	0.61	0.27	0.03	0.00	0.70	0.30
4.10 Data transport method(s) and specifications have been defined	0.12	0.36	0.48	0.03	0.00	0.48	0.52
4.11 A rigorous requirements inspection process has been implemented	0.27	0.30	0.21	0.21	0.00	0.58	0.42

stage the system truly begins to take shape as an interconnected system and the functionality of the parts can be seen from a system perspective. In many cases, development projects tend to bypass or minimize this stage because of time or funding constraints. However, the lack of upfront planning comes back in the form of reduced or unintended functionality later in development that can lead to even larger time and resource hits. Only by completing a comprehensive mapping of the system early in development can the true magnitude of the task be understood and successfully planned for.

In looking back at the key identified elements of the semantic stage we see a clear flow mapped out by integration SMEs. By considering the fundamental components of an integration effort as the technologies, their identified linkage (e.g. I/O), and a representation of this relationship (e.g. architecture), then our data indicates this in *1.1 Principal integration technologies have been identified*, *2.2 Inputs/outputs for principal integration technologies are known, characterized and documented*, and *3.4 High-level system interface diagrams have been completed*. This progression is in keeping with the best practices laid out by numerous studies and system engineering guides and reflects a steady evolution of knowledge from the time that the components required are identified until a formal architecture is developed.

Table 12
IRL 5 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 5 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
5.1 An Interface Control Plan has been implemented (i.e., Interface Control Document created, Interface Control Working Group formed, etc.)	0.33	0.58	0.06	0.00	0.03	0.91	0.06
5.2 Integration risk assessments are ongoing	0.06	0.48	0.45	0.00	0.00	0.55	0.45
5.3 Integration risk mitigation strategies are being implemented & risks retired	0.03	0.52	0.39	0.06	0.00	0.55	0.45
5.4 System interface requirements specification has been drafted	0.39	0.36	0.24	0.00	0.00	0.76	0.24
5.5 External interfaces are well defined (e.g., source, data formats, structure, content, method of support, etc.)	0.27	0.55	0.18	0.00	0.00	0.82	0.18
5.6 Functionality of integrated configuration items (modules/functions/assemblies) has been successfully demonstrated in a laboratory/synthetic environment	0.21	0.52	0.27	0.00	0.00	0.73	0.27
5.7 The Systems Engineering Management Plan addresses integration and the associated interfaces	0.15	0.18	0.33	0.12	0.21	0.33	0.45
5.8 Integration test metrics for end-to-end testing have been defined	0.12	0.33	0.52	0.03	0.00	0.45	0.55
5.9 Integration technology data has been successfully modeled and simulation	0.06	0.67	0.18	0.09	0.00	0.73	0.27

4.2. Syntactic (IRL 4-7)

For IRLs 4 and 5, we see less clarity in the identification of IRL decision criteria with more ambiguity in what is most important. This is not too different from what has been described with TRL, in that the transition from TRL 3 to 4 is the most ill defined and difficult to determine [1]. A great deal of this uncertainty can be attributed to the broad array of activities taking place at this stage of development, many of which are highly dependent on the type of project being worked. Depending on the complexity, goals, and knowledge base of work being undertaken, key activities could vary dramatically. For an effort that is truly revolutionary and untested, significantly more attention would be spent on risk analysis, quality assurance, and modeling and simulation whereas projects involving work of a more known quantity would be justified in focusing less in these areas and instead leveraging the significant number of lessons learned from projects that have gone before them. As reflected by the tightly grouped results, all criteria are important considerations and should receive attention while those that are of greatest impact to the project should be identified via careful consideration of project needs, priorities and risks.

For IRL 6 and 7 we begin to see more clarity again as IRL 6 shows two decision criteria as being critical. This is reflective of the common string of development activities that being to again reign supreme independent of the type of project being worked. As the technology elements are brought

Table 13
IRL 6 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 6 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
6.1 Cross technology issue measurement and performance characteristic validations completed	0.27	0.39	0.33	0.00	0.00	0.67	0.33
6.2 Software components (operating system, middleware, applications) loaded onto subassemblies	0.45	0.33	0.12	0.03	0.06	0.79	0.15
6.3 Individual modules tested to verify that the module components (functions) work together	0.48	0.42	0.09	0.00	0.00	0.91	0.09
6.4 Interface control process and document have stabilized	0.09	0.48	0.36	0.03	0.03	0.58	0.39
6.5 Integrated system demonstrations have been successfully completed	0.21	0.58	0.15	0.06	0.00	0.79	0.21
6.6 Logistics systems are in place to support Integration	0.12	0.42	0.27	0.18	0.00	0.55	0.45
6.7 Test environment readiness assessment completed successfully	0.06	0.52	0.33	0.06	0.03	0.58	0.39
6.8 Data transmission tests completed successfully	0.18	0.64	0.06	0.06	0.06	0.82	0.12

Table 14
IRL 7 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 7 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
7.1 End-to-end Functionality of Systems Integration has been successfully demonstrated	0.61	0.18	0.21	0.00	0.00	0.79	0.21
7.2 Each system/software interface tested individually under stressed and anomalous conditions	0.33	0.55	0.12	0.00	0.00	0.88	0.12
7.3 Fully integrated prototype demonstrated in actual or simulated operational environment	0.42	0.45	0.09	0.03	0.00	0.88	0.12
7.4 Information control data content verified in system	0.24	0.55	0.18	0.00	0.03	0.79	0.18
7.5 Interface, Data, and Functional Verification	0.33	0.55	0.09	0.03	0.00	0.88	0.12
7.6 Corrective actions planned and implemented	0.15	0.48	0.27	0.09	0.00	0.64	0.36

together and the interfaces are fully defined and made to function an urgent need to initiate testing comes about for development efforts. In order to mitigate the difficulty of large system testing later in the development cycle it is viewed as a critical step that smaller elements or modules of functionality be flexed in order to assess the completeness of their integration. (see *6.3 Individual modules tested to verify that the module components* (*functions*) *work together*). This then evolves as these modules are further integrated into an overarching functional system for continued testing. For IRL 7 we indicate

Table 15
IRL 8 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 8 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
8.1 All integrated systems able to meet overall system requirements in an operational environment	0.85	0.12	0.03	0.00	0.00	0.97	0.03
8.2 System interfaces qualified and functioning correctly in an operational environment	0.61	0.36	0.03	0.00	0.00	0.97	0.03
8.3 Integration testing closed out with test results, anomalies, deficiencies, and corrective actions documented	0.39	0.52	0.09	0.00	0.00	0.91	0.09
8.4 Components are form, fit, and function compatible with operational system	0.42	0.48	0.06	0.03	0.00	0.91	0.09
8.5 System is form, fit, and function design for intended application and operational environment	0.42	0.45	0.09	0.03	0.00	0.88	0.12
8.6 Interface control process has been completed/closed-out	0.24	0.45	0.24	0.06	0.00	0.70	0.30
8.7 Final architecture diagrams have been submitted	0.36	0.12	0.42	0.09	0.00	0.48	0.52
8.8 Effectiveness of corrective actions taken to close-out principal design requirments has been demonstrated	0.24	0.48	0.24	0.03	0.00	0.73	0.27
8.9 Data transmission errors are known, characterized and recorded	0.36	0.33	0.21	0.09	0.00	0.70	0.30
8.10 Data links are being effectively managed and process improvements have been initiated	0.18	0.52	0.27	0.03	0.00	0.70	0.30

that end-to-end testing (see *7.1 End-to-end Functionality of Systems Integration has been successfully demonstrated*) is critical before moving to our next phase – Pragmatic (or operation). We believe this is consistent with prescribed system development phases [10]. Unfortunately, many programs see this critical end-to-end testing phase squeezed in a race to field a capability or stay on schedule. In order to successfully pass the IRL 7 stage, however, it is essential that a complete and thorough test of the newly developed system be conducted to prove that the functionality is as desired and that the reliability of the system is suitable for operation.

4.3. Pragmatic (IRL 8–9)

Since Pragmatic addresses the operational context of the integration, it is not surprising that decision criteria such as meeting requirements become paramount. At this phase of system maturation, developmental and operational testing activities are used to determine the degree to which the system meets the requirements outlined for the effort at project initiation (*8.1 All integrated systems able to meet overall system requirements in an operational environment; 8.2 System interfaces qualified and functioning correctly in an operational environment*).

These activities ensure that the system can function fully not only in a laboratory or experimental situation but in a realistic environment where many factors cannot be readily controlled or anticipated.

Table 16
IRL 9 decision criteria and criticality assessment

	Relative frequency (RF); $n = 33$					Cumulative RF	
IRL 9 decision criteria	Critical	Essential	Enhancing	Desirable	N/A	Critical Essential	Enhancing Desirable
9.1 Fully integrated system has demonstrated operational effectiveness and suitability in its intended or a representative operational environment	0.82	0.09	0.09	0.00	0.00	0.91	0.09
9.2 Interface failures/failure rates have been fully characterized and are consistent with user requirements	0.64	0.27	0.06	0.03	0.00	0.91	0.09
9.3 Lifecycle costs are consistent with user requirements and lifecycle cost improvement initiatives have been initiated	0.24	0.42	0.21	0.09	0.03	0.67	0.30

Unfortunately, in recent years there has been a trend towards the waiving of requirements not attained by this system late in the design cycle. Instead of ensuring the that system is fully capable, the symptoms of a dysfunctional integration process often result in the acceptance of a system that is of a lesser capability than was desired or needed. This is one of the shortcomings that the development of a rigorous integration scale is intended to mitigate. The final stage of integration maturity, IRL 9, can only be attained after a system has truly been flexed by the operator and is independent of the type of project undertaken. The important criteria principally take into account quantification and demonstration in operational environment (*9.1 Fully integrated system has demonstrated operational effectiveness and suitability in its intended or a representative operational environment*), and failure rate characterization (*9.2 Interface failures/failure rates have been fully characterized and are consistent with user requirements*) all of which were rated high by SMEs. At this final stage the fruits of a successful system maturation process can be seen through a highly functional capability with robust reliability. An inability to achieve satisfactory results should be prevented through the proper application and tracking of Technology and Integration Readiness Levels.

4.4. Summary and future research

Theoretically the two activities of technology development and integration could be represented on a linear plane. Although, we do not contend that these developments are parallel paths, thus there is a dynamic, non-linear causality akin to the embedded systems engineering life cycle (or "V within the V within the V...”). We presented IRL as a management tool built from the 7-layer Open Systems Interconnect model used to build computer networks. IRL has been designed to be used in conjunction with an established technology metric, i.e. TRL, to provide a system-level readiness assessment. Ideally, the two of these metrics used in conjunction with each other can provide a common language that can improve organizational communication of scientists, engineers, management, and any other relevant stakeholders on integration within documented systems engineering guidance (e.g. [10,16,17,28]). We complemented the IRL with a checklist that would allow for the removal of some of the subjectivity that exist in many of the maturity metrics.

This said, IRL is not a complete solution to integration maturity determination; it is although a tool that increases the stakeholder communication, something that has proven to be critical in all the case studies presented [21,24,27,32]. Yet, the case studies indicate:

- IRL lacks the ability to assess criticality and R&D effort
- IRL assessment of complex, net-centric systems requires a more quantitative algorithm to reduce multiple integrations to a single assessment
- IRL does not evaluate cost and schedule

Additionally, for this study the participants were asked to assess the criticality of each IRL metric within the context of the IRL they were listed under rather than being allowed to identify metrics that they considered useful in assessing the IRL as defined. In other words, participants were given a "canned" list of metrics and a "fixed" context (i.e., the IRL construct and the specific IRL that a set of metrics was assigned to). Therefore, it is recommended that additional work be conducted (perhaps via multiple working groups comprised of seasoned practitioners or SMEs) to review and modify the current list of IRL metrics while using the criticality assessment as a baseline. This effort should address two aspects of the IRL checklist: the metrics themselves and the weight that should be assigned to each based on criticality data. Additionally, the issue of whether or not the integration type is an important factor concerning how an IRL is determined needs to be examined.

Finally, integration is a complex topic and the respondents may have been biased by the type of integration experience they have had (i.e., software, hardware, software and hardware, etc.); the wording of each IRL metric may have been interpreted differently by the participants; and some decision criteria may belong within a different IRL scale, thereby altering its criticality.

IRL is not without its limitations, and it is these issues that must be the focus of future work. One example resulting from the case study assessment is that IRL is able to uncover integration maturity concerns even if the TRLs of the integrating technologies are high. This was not the case with all of the metrics; however, ITI is able to factor in R&D effort and technology criticality into the assessment, assuming technology maturity is still low. It may be that a hybrid metric that uses both IRL and ITI is the solution to this situation.

Future work includes:

- Apply IRL to systems under development to better understand how the metric works in practicality.
- What is the impact of emergent behavior in system integration, and how does IRL handle this?
- At what level of the system architecture does IRL get applied?
- What are the dynamics of progressing through the IRL scale?
- Incorporate ITI assessment into the TRL/IRL assessment process:
  - III (Integrated Integration Index)?
  - In a net-centric architecture does the simple summarization in ITI still apply?
- Determine how it is possible to simplify complex architectures to represent system-level readiness:
  - System Readiness Levels (SRL) = f(TRL,IRL)? [34]
  - What is the value of a SRL?
  - If system architectures can be represented as graphs, how can graph theory be applied to determine a SRL as a function of TRL, IRL, and possibly ITI?
  - How can UML/SysML be applied to create a framework for system maturity assessment?; Can a new "view" be created within SysML specifically to address TRL, IRL, and SRL?

References

[1] M. Austin, J. Zakar, D. York, L. Pettersen and E. Duff, *A Systems Approach to the Transition of Emergent Technologies into Operational Systems – Herding the Cats, the Road to Euphoria and Planning for Success* International Conference of the International Council on Systems Engineering, INCOSE, Netherlands, 2008.

[2] J.S. Beasley, *Networking*, Pearson Education, Inc., Upper Saddle River, NJ, 2004.

[3] J.W. Bilbro, *A Suite of Tools for Technology Assessment*, Technology Maturity Conference: Multi-Dimensional Assessment of Technology Maturity, AFRL, Virginia Beach, VA, 2007.

[4] R.T. Brooks, and A. Sage, System of systems integration and test, *Information Knowledge Systems Management* **5**(2005/2006), 261–280.

[5] D.M. Buede, *The Engineering Design of Systems*, John Wiley & Sons, New York, 2000.

[6] J. Connelly, K. Daues, R.K. Howard and L. Toups, *Definition and Development of Habitation Readiness Level (HRL) for Planetary Surface Habitats*, 10th Biennial International Conference on Engineering, Construction, and Operations in Challenging Environments, Earth and Space, 2006, 81.

[7] D. Cundiff, *Manufacturing Readiness Levels (MRL)*, Unpublished Whitepaper, 2003.

[8] DoD, *Levels of Information Systems Interoperability*, 1998.

[9] DoD, *Technology Readiness Assessment (TRA) Deskbook*. in: DUSD(S&T), (Ed.), Department of Defense, 2005.

[10] DoD, *Operation of the Defense Acquisition System*, Instruction 5000.02 Department of Defense, Washington, DC, 2008.

[11] T. Dowling and T. Pardoe, in: *TIMPA – Technology Insertion Metrics*, M.o. Defense, ed., QINETIQ, 2005, p. 60.

[12] J. Fang, S. Hu and Y. Han, *A Service Interoperability Assessment Mode l for Service Composition*, IEEE International Conference on Services Computing (SCC'04), 2004.

[13] GAO, *Best Practices: Better Management of Technology Development Can Improve Weapon System Outcomes*. in: GAO, (Ed.), 1999.

[14] C.P. Graettinger, S. Garcia, J. Siviy, R.J. Schenk and P.J.V. Syckle, *Using the "Technology Readiness Levels" Scale to Support Technology Management in the DoDs ATD/STO Enviornments*. in: SEI, (Ed.), Carnegie Mellon, 2002.

[15] M. Hobday, H. Rush and J. Tidd, Innovation in complex products and system, *Research Policy* **29**(7–8) (2000), 793–804.

[16] IEEE, *Systems and software engineering – System life cycle processes*, IEEE 15288, 2008.

[17] INCOSE, *INCOSE Systems Engineering Handbook*, Version 3, International Council on Systems Engineering, 2007.

[18] ISO, *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, ISO/IEC 7498-1, 1994, 1–68.

[19] JPL, *Mars Global Surveyor: Mission Operations Specifications: Volume 5, Part 1*, California Institute of Technology, 1996, 1–51.

[20] L.J. Lanzerotti, in: *Assessment of Options for Extending the Life of the Hubble Space Telescope: Final Report*, N.R. Council, ed., The National Academies Press, Washington D.C., 2005.

[21] J.L. Lions, *ARIANE 5: Flight 501 Failure, Report by the Inquiry Board*, Paris, 1996.

[22] J.C. Mankins, *Technology Readiness Levels*, NASA, 1995.

[23] J.C. Mankins, Approaches to Strategic Research and Technology (R&T) Analysis and Road Mapping. *Acta Astronautica* **51**(1–9) (2002), 3–21.

[24] J.J. Mattice, in: *Hubble Space Telescope: Systems Engineering Case Study*, C.f.S. Engineering, ed., Air Force Institute of Technology, 2005.

[25] D.E. Mosher, Understanding the Extraordinary Cost Growth of Missile Defense, *Arms Control Today*(2000), 9–15.

[26] S. Nambisan, Complementary product integration by high-technology new ventures: The role of initial technology strategy, *Management Science* **48**(3) (2002), 382–398.

[27] NASA, *Mars Climate Orbiter, Mishap Investigation Board: Phase I Report*, 1999.

[28] NASA, *NASA Systems Engineering Handbook*, NASA/SP-2007-6105, National Aeronautics and Space Administration, Washington, DC, 2007.

[29] E.G. Nilsson, E.K. Nordhagen and G. Oftedal, *Aspects of Systems Integration*, 1st International System Integration, 1990, 434–443.

[30] A.M. Orme, H. Yao and L.H. Etzkorn, Coupling Metrics for Ontology-Based Systems, *IEEE Software* (2006), 102–108.

[31] A.M. Orme, H. Yao and L.H. Etzkorn, Indicating ontology data quality, stability, and completeness throughout ontology evolution, *Journal of Software Maintenance and Evolution* **19**(1) (2007), 49–75.

[32] S.R. Sadin, F.P. Povinelli and R. Rosen, The NASA Technology Push Towards Future Space Mission Systems, *Acta Astronautica* **20**(1989), 73–77.

[33] P.A. Sandborn, T.E. Herald, J. Houston and P. Singh, Optimum Technology Insertion Into Systems Based on the Assessment of Viability, *IEEE Transactions on Components and Packaging Technologies* **26**(4) (2003), 734–738.

[34] B. Sauser, J. Ramirez-Marquez, D. Henry and D. DiMarzio, A System Maturity Index for the Systems Engineering Life Cycle, *International Journal of Industrial and Systems Engineering* **3**(6) (2008), 673–691.

[35] B. Sauser, D. Verma, J. Ramirez-Marquez and R. Gove, *From TRL to SRL: The Concept of System Readiness Levels*, Conference on Systems Engineering Research (CSER), Los Angeles, CA USA, 2006.
[36] R. Shishko, D.H. Ebbeler and G. Fox, Nasa Technology Assessment Using Real Options Valuation, *Systems Engineering 7*(1) (2003), 1–12.
[37] J. Smith, *An Alternative to Technology Readiness Levels for Non-Developmental Item* (*NDI*) *Software*, Carnegie Mellon, Pittsburgh, PA, 2004.
[38] R. Valerdi and R.J. Kohl, *An Approach to Technology Risk Managment*, Engineering Systems Division Symposium, MIT, Cambridge, MA, 2004.
[39] R.J. Watts and A.L. Porter, R&D cluster quality measures and technology maturity, *Technological Forecasting & Social Change* **70**(2003), 735–758.

Brian Sauser holds a B.S. from Texas A&M University in Agricultural Development with an emphasis in Horticulture Technology, a M.S. from Rutgers, The State University of New Jersey in Bioresource Engineering, and a Ph.D. from Stevens Institute of Technology in Project Management. He is currently an Assistant Professor in the School of Systems & Enterprises at Stevens Institute of Technology. Before joining Stevens in 2005, he spent more than 12 years working in government, industry, and academia both as a researcher/engineer and director of programs. His research interest is in the management of complex systems. This includes system and enterprise maturity assessment and the advancement of a foundational science of systems thinking. He is currently the Director of the Systems Development and Maturity Laboratory (http://www.systems-development-maturity.com), which seeks to advance the state of knowledge and practice in systems maturity assessment. He teaches courses in Project Management of Complex Systems, Designing and Managing the Development Enterprise, and Systems Thinking. In addition, he is a National Aeronautics and Space Administration Faculty Fellow, Editor-in-Chief of the *Systems Research Forum*, an Associate Editor of the *IEEE Systems Journal*, and the Associate Editor of the *ICST Transactions on Systomics, Cybernetics, and e-Culture*.

Ryan Gove is a Systems Engineer at Lockheed Martin Simulation, Training, and Support in Orlando Florida where he supports Turn-Key Flight Training development. Ryan received his B.S. in Computer Engineering from University of Delaware and his M.E. in Systems Engineering from Stevens Institute of Technology. He is currently pursuing a PhD in Systems Engineering at Stevens. His current research interests include system maturity assessment frameworks as applied to the management of complex system design, integration, and operation. He is also interested in system of systems management, graph theory, and UML/SysML applications.

Eric Forbes is a marketing and business development specialist with 3M's Aerospace and Aircraft Maintenance Department focusing on the application of advanced adhesive, acoustic, and thermal technologies to the aerospace market. During the writing of this paper, Eric worked as a systems engineer with the Northrop Grumman Corporation's Littoral Combat Ship Mission Package Integrator program and was part of the joint industry, government and academia development of the System Readiness Level methodology. His previous work within Northrop Grumman included research and development, systems engineering, and business development activities for a wide cross section of missile systems and C4ISR projects. Forbes earned a Bachelor's degree in Aeronautical and Astronautical Engineering from the University of Washington and a Master's degree in Aerospace Engineering from the Georgia Institute of Technology.

Jose Emmanuel Ramirez-Marquez received the M.Sc. degree in statistics from the Universidad Nacional Autonoma de Mexico, Mexico City, Mexico, and the M.Sc. and Ph.D. degrees in industrial engineering from Rutgers University, New Brunswick, NJ. He is currently an Assistant Professor in the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ. He was a Fulbright Scholar. His current research interests include the reliability analysis and optimization of complex systems, the development of mathematical models for sensor network operational effectiveness, and the development of evolutionary optimization algorithms. He has conducted funded research for both private industry and government. He is the author or coauthor of more than 50 refereed manuscripts related to these areas in technical journals, book chapters, conference proceedings, and industry reports. Ramirez-Marquez has presented his research indings both nationally and internationally in conferences such as Institute for Operations Research and Management Science (INFORMS), Industrial Engineering Research Conference (IERC), Applied Reliability Symposium (ARSym), and European Safety and Reliability (ESREL). He is the Director of the Quality Control and Reliability (QCRE) Division Board of the Institute of Industrial Engineers and is a member of the Technical Committee on System Reliability for the ESRA.