A Systems Integration Framework for Process Analysis and Improvement

Rashmi Jain,* Anithashree Chandrasekaran, and Ozgur Erol

Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030

Received 16 July 2007; Revised 7 October 2008; Accepted 5 May 2009, after one or more revisions
Published online in Wiley InterScience (www.interscience.wiley.com)
DOI 10.1002/sys.20148

ABSTRACT

Systems Integration (SI) is an important element of Systems Engineering. It involves the integration of hardware, software, products, services, processes, and humans. The ever-increasing scale of complexity of systems and its impact on the business requires that we revisit the processes involved in the development and integration of a system. This paper proposes a Systems Integration Process Model (SIPM) based on a comprehensive lifecycle view of systems integration. As part of the ongoing SI research at Stevens Institute of Technology, the authors have developed a Systems Integration Framework (SIF) which incorporates the relevant aspects of integration from a lifecycle perspective and sets a foundation to an end-to-end approach to SI. Our end-to-end approach focuses on how integration issues can be addressed up-front to minimize integration related complexities and challenges later on in the system engineering process. This paper discusses the merits and benefits of applying the SIPM to evaluate and improve current SI processes in organizations. The paper provides, in addition to an overview of the SI framework, the activities included in the model. The model was pilot tested to evaluate the SI processes at a government agency. The results were used to provide recommendations for SI process reengineering. © 2009 Wiley Periodicals, Inc. Syst Eng

Key words: system integration; verification and validation; interface; COTS; legacy; integration process; integration activities

I. INTRODUCTION

The challenge of addressing systems integration complexity and qualifying a system for deployment or implementation demands new innovative methods for qualification, integration, and testing. While researching for methodologies for effective systems integration, we realized that, in order to develop a new methodology, a clear and concise understanding of the processes and activities involved in SI is much needed.

Systems integration activities are highly coupled and supported by activities across a system lifecycle. This comprehensive view of SI integrates both the products and the processes to achieve the required objectives. A SI framework approach was taken to address the challenge. Based on this framework, five process areas of SI were identified, and activities under each process area were studied. The first iteration of developing a SIPM used the literature review of the existing and available process models in SI and Systems Engineering (SE) as its foundation. This model was then refined by the study of industry best practices for SI and SE. This model was then piloted at a government agency to

*Author to whom all correspondence should be addressed (e-mail: rashmi.jain@stevens.edu; Anithashree.chandrasekaran@stevens.edu; oerol@stevens.edu).
evaluate its current integration process effectiveness and propose gaps and recommendations to improve the effectiveness.

The next several sections will discuss the details of SI Framework and the lifecycle view, SI processes model, and related activities, and the results from the application of the model at a government agency.

2. SYSTEMS INTEGRATION AND SYSTEMS ENGINEERING

Systems Integration is an important element of Systems Engineering which involves the integration of hardware, software, products, services, business processes, and humans [Grady, 1994; Jain, 2007]. From a process perspective, the systems integration process creates the links within the Systems Engineering process from requirements collection to verification and validation and ultimately to operation of the system. It would be accurate to state that the SI process and activities occur throughout the SE lifecycle. SI should be implemented from the beginning and throughout the system development rather than being implemented only as a “downstream” event.

The existing standards, models, and guidelines of systems engineering and software engineering address systems integration issues partially. They tend to view systems integration from a perspective of integrating physical components. These standards and models lack a holistic end-to-end approach to SI. For example, 15288:2002 [ISO/IEC, 2002] systems integration process activities include:

- Define an assembly sequence and strategy that minimizes systems integration risks.
- Identify the constraints on the design arising from the integration strategy.
- Obtain integration enabling systems and specified materials according to the defined integration procedures.
- Obtain system elements in accordance with agreed schedules.
- Assure that the system elements have been verified against acceptance criteria specified in an agreement.
- Integrate system elements in accordance with applicable interface control descriptions and defined assembly procedures.
- Use the specified integration facilities.
- Record integration information in an appropriate database [ISO/IEC, 2002].

This definition of systems integration does not include the important issues of systems integration such as interoperability, interface control and management, business process integration, and traceability. Due to the emerging SE challenges and the increasing importance of systems integration, the need for a holistic approach to Systems Integration has become critical.

3. SYSTEMS INTEGRATION FRAMEWORK (SIF)

As a part of the ongoing systems integration research at Stevens Institute of Technology, a Systems Integration Framework (Fig. 1) was developed based on the literature review and evaluation of systems integration processes and models. This framework was developed as a baseline to identify a comprehensive set of end-to-end activities that may constitute and define the scope of systems integration processes. The details on this framework can be obtained from Jain, Erol, and Chandrasekaran [2009].

This Systems Integration Framework illustrates a comprehensive end-to-end systems integration approach. This approach is based on the premise that integration occurs...
throughout the lifecycle of a system. Systems integration is not a one-time activity. Our current focus of SI processes and activities looks at the following three dimensions of SI interactions as illustrated in the SI framework. (1) Interoperability is a prerequisite to achieving successful systems integration. If the subsystems or systems are not interoperable, they cannot be integrated. Standards or other forms of guidelines—international, national, or industry-specific—provide an opportunity to design systems which can be “open” and simple to integrate. On the other hand, these guidelines can also constrain the design by requiring additional design elements that may lead to more complex systems integration. (2) The other processes of system development lifecycle such as requirements management, interface control and management, testing, verification and validation, and configuration management should include and address SI issues. (3) Legacy systems integration and COTS integration issues are inevitable in integration of most systems. Interoperability and interface management issues of legacy and COTS systems affect the overall success of systems integration to a significant extent.

Due to the shift within the SE community towards considering “enterprises” as systems [Rouse, 2005zaq;1; Carlock and Fenton, 2001zaq;1; Kosanke et al., 1999zaq;1], Enterprise Integration (EI) has become an important aspect of SI. EI is related to integrating technology, processes, and people to facilitate a greater flow of information and effective decision making across the enterprise with the goal of improving efficiency and competitive advantage. Integration of business systems, or for that matter any system, requires a good understanding of business processes and business process analysis. Systems are designed, developed, and engineered to support the business processes within an enterprise and alignment of processes and systems is crucial to better meet the needs of stakeholders for whom the systems are built. This approach sets our foundation of the end-to-end (life cycle) approach to SI. Therefore, business process integration (BPI) is an important element of our SI Framework. An understanding of the concept of a business process and the need to conduct integrated business process analysis is a prerequisite for systems integration. Though the authors believe both EI and BPI are essential for an end-to-end integration the scope of the research study of the agency’s SI process does not include them due to programmatic limitations.

4. SYSTEMS INTEGRATION LIFECYCLE AND PROCESSES

Extending the framework, we define the Systems Integration process as a set of activities that transforms the stakeholder requirements into an operational system by unifying the process components and product components into a whole while ensuring compliance to the specified levels of component operations and interoperability. The scope and objectives of systems integration should be clearly identified to ensure that new systems and components are able to work with the existing systems and components. This can be achieved by applying [Mische, 1998] four states of systems integration:

- Interconnectivity is the initial and the fundamental state in the systems integration. It requires all new and existing system components and equipment to connect and work together.
- Interoperability means that all interconnected information system components and equipment should be able to function and interact with each other. Interoperability is considered as the key state of systems integration.
- Semantic consistency refers to the concern of consistency at data level.
- Convergent integration involves the amalgamation of components and technology with business processes, people, skills, and knowledge.

A lifecycle view of SI will help us in understanding the context of SI and its scope across system engineering life cycle phases. It also helps in identifying and addressing SI issues as and when they happen throughout the system development, implementation, and operation.

Figure 2 shows how the five Systems integration subprocesses fit within the Systems Engineering Process. These processes are embedded within the SE lifecycle processes in places where SI activities become critical and important. The SI lifecycle has five high level subprocesses. These subprocesses are identified as Integration Requirements; Integration Architecture; Integration Planning; Integration Implementation; and Integration Validation and Verification as shown in Figure 2.

Once a conceptual design for a system is chosen and all operational scenarios (use cases) to understand the context are analyzed, the broader category of stakeholder requirements are then refined and derived to form system requirements or specifications. During this phase of SE life cycle, it is critical to identify requirements that will impact systems integration. These requirements termed as integration requirements address the required level of integration and quality. These integration requirements fall under the categories of compliance, interoperability, qualification, COTS (commercial off-the-shelf), and legacy. This subprocess of identifying, refining, deriving, and managing the above categories of requirements is called Integration Requirements subprocess.

The integration requirements are then used to develop the design (architecture) to address the integration issues such as COTS, legacy, interfaces, testability, qualification, and compliance. These design decisions are based on the functional and physical architecture of the system. Tradeoff decisions are based on these architectures and requirements. This subprocess of developing architectures to integrate the system as a whole using tradeoffs and decision making is called the Integration Architecture subprocess.

Based on the optimized system architecture design, plans are developed to manage and implement interfaces and tests. The plans are based on the interface architecture and qualification architecture. The planning decisions are made concurrently based on the system model optimization sub-process. This subprocess of harmonizing the interface management, verification, and validation of the integrated subsystems and the system with its environment is called Integration Planning subprocess.
Once the detailed designs are implemented and subsystems are built, verification and validation of the subsystems is done as per the requirements, architecture, and the plan. These subsystems are then integrated and the process of verification and validation occurs to determine the compliance and qualification of the integrated system. This process iterates until the completed system integrates with its operational environment. The subprocess of integrating the subsystems and system as a whole based on the architectures is called Integration Implementation subprocess. The subprocess of testing and qualifying the subsystems and the integrated system is called Integration Verification and Validation subprocess.

Based on the system development lifecycle, these subprocesses can feed into each other and iterate to mitigate and manage the unknown factors of the system and optimize integration effectiveness. These five subprocesses of systems integration form the five fundamental processes of systems integration process model. SIPM identifies all the subprocesses that are critical for systems integration. It further explains the interaction of activities within each subprocess, and between the five subprocesses. The following section discusses the activities in each subprocess.

5. SYSTEMS INTEGRATION PROCESS MODEL (SIPM)

SIPM shows how each of the sub-processes interacts and how the process flow occurs in each. For the purpose of clarity and readability, the process model is shown in four process flow diagrams (Figs. 3, 4, 6, and 7). The interconnections between the processes are shown in a rounded rectangle with the activity number listed. The activity numbers are shown along with the activity name in each process flow. SIPM has five subprocesses and 45 activities. This model was developed based on literature reviews on systems integration, standards, and integration best practices. The following subsections provide a brief discussion on each subprocess and their process flows.

The process flows are used only for the understanding of the dependence between each activity in systems integration. The SI process execution sequence or concurrence is not shown in the process flows as included in this paper. However, the direction and nature of process flows forms the basis of traceability between all the activities related to each of the processes (artifacts, resources, etc.). Traceability facilitates validation in each subprocess. Validation helps us determine that these systems integration processes have produced the right design and related artifacts. The process flows included in this paper do not explicitly show the traceability across them and the validation at the end of each subprocess.

5.1. Integration Requirements Subprocess

Integration Requirements is one of the most important subprocesses of the systems integration process. The process flow of the requirements sub-process is shown in Figure 3. In order
to avoid issues and surprises during the implementation, it is important to elicit system requirements with a systems integration focus early on. The delays in product development are mainly attributed to errors and rework that result due to the errors during the final phases. The effort and money spent on these errors and fixes are huge [McConnell, 1996]. These errors could have been avoided with early consideration of integration issues. It is a good practice to identify systems integration stakeholders and gather requirements from them early in the SE life cycle. The systems integration process begins with these requirements subprocess activities. In general, systems integration requirements fall under six major categories or types

5.1.1. Interoperability Requirements
Interoperability is the ability of systems to provide services to and accept services from other systems, and to use the services so exchanged to enable them to operate effectively together. Interoperability can be achieved by designing and building systems against a defined interoperability requirement, and maintaining that interoperability throughout as the system changes and upgrades through configuration management, and testing for interoperability against those requirements. Interoperability Requirements address or help address an operationally recognizable activity or sequence of activities that has a definable starting action, a definable concluding action, and which involves the exchange of items like data, information, material, or energy between two or more systems (subsystems or platforms). Such an exchange may be interactive and may involve the use of more than one transfer medium; however, the information content on all transfer media must be definable. These requirements are related to an operational capability. In most cases, few interoperability requirements are identified and interoperability often becomes an issue when a system is deployed.

5.1.2. Interface Requirements
Interface is a connection resource for linking to another system’s interface or to another system’s component. Interface is the functional and physical characteristics required to exist at a common boundary or connection between systems or items. Interface requirements must address total system performance, the fidelity of the interface, and any system requirements meant to constrain interface design [Buede, 2000]. Interface requirements are statements on the interface designs, protocols used, data formats, entity relationships, and processing rules. They define the interfaces and their inputs and outputs. Interface Interactions between elements [Pimmler and Eppinger, 1994] are:

1. Spatial: A spatial-type interaction identifies needs for adjacency or orientation between two elements.
2. Energy: An energy-type interaction identifies needs for energy transfer between two elements.
3. Information: An information-type interaction identifies needs for information or signal exchange between two elements.
4. Material: A material-type interaction identifies needs for materials exchange between two elements.

In cases where systems integration is highly linked to the organizational planning and operations and in cases where end-user computing is involved, organizational interfaces become critical. Organizational interfaces are the common boundaries between user, system, and organization. The nature of the interfaces can be procedures, data, personnel, hardware, and software [Trauth and Cole, 1992]. Organizational interfaces mainly include communication between user/system, and organization and its subunits, user/system, and organizational vendors and subcontractors, organization/team coordination, training, documentation, support and services, business alignment planning, organizational innovation [Beise, 1994; Trauth and Cole, 1992]. The quality of such interfaces also determines system effectiveness [Beise, 1994]. Implications on system design and development due to organizational interfaces in terms of organizational forms of support are discussed in Trauth and Cole [1992] and in other literature discussed in the former.

Interface requirements for all external and internal interfaces are gathered in the early phases of systems integration. Interface requirements gathering and analysis help in understanding the critical variables of all internal and external interfaces and their predicted variations but only to some extent—those that can be foreseeable. However, in the operational environment there is always a challenge to deal with the impact of interaction of these critical variables that was not foreseen. An upfront and early-on focus on interface requirements and architecture helps in addressing these issues. Interface requirements help in architecting interfaces to achieve robustness and specified level of “ilities.”

5.1.3. Qualification/Test Requirements
Qualification requirements address the needs to qualify the system as being designed right, the right system, and an acceptable system [Buede, 2000]. Qualification is the process of verifying and validating the system design and then obtaining the stakeholder’s acceptance of the design. Qualification is associated with testing, acceptance, verification, and validation. The four elements of the qualification requirements are (i) observance—how the expected qualification data for each input/output and system-wide requirements will be obtained, that is, test, analysis and simulation, inspection, or demonstration; (ii) verification plan—how the qualification data will be used to determine that the real systems conforms to the design that is developed; (iii) validation plan—how the qualification data will be used to determine that the real system complies with the originating requirements; and (iv) acceptance plan—how the qualification data will be used to determine that the real system is acceptable to the stakeholders.

5.1.4. Operational Readiness Requirements
Operational readiness requirements address requirements that help to ensure that the solution can be correctly deployed within the enterprise or the operational environments. These requirements identify the entire roll out procedures and programs that are required and production support to install or deploy the system or release in its operational environments. These requirements are obtained from ConOps and service level agreements of the system. ConOps describes the result
of the system conceptual analysis process. They include all of
the information needed to describe the users’ needs, goals,
expectations, operational environment, processes, and char-
acteristics for the system under consideration. Operational
readiness requirements focus on the sections of ConOps that
include modes of operation for the proposed system, user
classes and user characteristics, operational features of the
proposed system, and operational scenarios for each opera-
tional mode defining the system’s interaction with other sys-
tems. These requirements ensure that all operating scenarios
can be supported by the proposed system.

5.1.5. Integration Technology Requirements
Integration technology requirements address systems integra-
tion with technical solutions. They address details on mecha-
nisms that need to be implemented which will allow the
transfer of data between systems, and mechanisms for initiat-
ing actions in other systems. Integration technology require-
ments address feasibility, availability, and relevance of
technical solutions for optimal integration. In most cases
integration technology focuses on the interconnectivity aspect
of the comprehensive systems integration. Some examples of
integration technology are data transfer, transport services,
file transfer protocol, document protocols, remote procedure
calls, etc.

5.1.6. Standards, Guidelines, and Recommendations
Requirements
Standards, guidelines, and recommendations requirements
are the constraining requirements that have to be complied
with for various reasons specific to each system and its
domain, for example, security and safety requirements. These
standards, guidelines, and recommendations can be organiza-
tional specific, technology specific, domain specific, audit
and regulatory requirements, and international standards for
quality, safety, and security. These include all formal, de jure,
and de facto standards relevant to the technology and the
proposed system. In most cases these requirements are used
to achieve interoperability, interchangeability, and portability.
Systems tend to have longer operational life if they are based
on universally accepted standards and have been time-tested.
These requirements can be both functional and nonfunctional.

The above six categories of Integration Requirements are
gathered from the stakeholders and are further analyzed and
refined as shown in Figure 3. The derived requirements are
requirements that help us better understand and support these
originating requirements (requirements obtained from stake-
holders). They are detailed system requirements for the cho-
gen system concept. These requirements need to be iterated
with the stakeholders of systems integration and signed off so
that they can be used as a baseline for systems integration
process. These requirements have to be documented well and
maintained with good configuration management (CM). Con-
figuration management plays a key role in achieving systems
integration effectiveness and is part of every subprocess of
integration. Four classic operational aspects of Configuration
Management (CM) are identification of the structure of the
product/process, control of their releases, accounting for their
status, and audit and review of their completeness.

Figure 3. Integration requirements activities flowchart. [Color fig-
ure can be viewed in the online issue, which is available at www.
interscience.wiley.com.]

5.2. Integration Architecture Subprocess
The integration requirements are addressed through the inte-
gration architecture which is embedded in both the physical
and functional architectures of the system. Figure 4 illustrates
the process model of the integration architecture subprocess.
Integration architecture includes legacy systems integration,
COTS integration, interface definition, control and manage-
ment, and qualification architecture. This section briefly ex-
plains each of these activities. [Jain, Erol, and
Chandrasekaran, 2009] provides detailed discussion on these
activities.

We identify the strategies for integrating legacy systems
during the conceptual design phase. These legacy systems are
usually mission/business critical, and should remain func-
tional at all times [Konstantas, 1996]. They fully satisfy
system functional requirements and support current business
functionality and have been thoroughly tested in the actual
operational environment. They are also coupled with the rest
of the operational infrastructure. The legacy systems provide
a set of design constraints which have to be identified and
defined. The limitations of legacy systems such as pollution,
embedded knowledge, poor lexicon, coupling, layered architectures, frequency of failures and breakdowns, obsolescence, and maintenance cost are discussed in detail in Bianchi et al. [2003] The requirements for legacy integration are gathered and analyzed. They are further derived to understand the required level of interoperability and interfaces for integrating the legacy system/subsystem. These requirements constrain the architectural decisions for the proposed system.

Once the system requirements are elicited and a baseline architecture is developed then build or buy related design decisions for subsystems and components are made. The commercial-off-the-shelf (COTS) subsystems or components are identified. A COTS item is one that is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor who retains the intellectual property rights; available in multiple, identical copies; and used without modification of the internals [OSD, 2002]. The requirements on the level of interoperability and interfaces for COTS integration are derived for further architectural refinement and decisions. A simultaneous COTS approach is recommended to addresses COTS integration issues. In this approach convergent decisions are made considering the following tradeoffs simultaneously [SEI, 2002]:

1. **Stakeholder needs and business processes**: Requirements (including quality attributes such as performance, security, and reliability), end-user business processes, business drivers, and operational environment.
2. **Marketplace**: Available and emerging COTS technology and products, nondevelopment items (NDI), and relevant standards.
3. **Architecture and Design**: Essential elements of the system and the relationships between them. Elements include structure, behavior, usage, functionality, performance, resilience, reuse, comprehensibility, economic and technologic constraints and tradeoffs, and aesthetic issues.
4. **Programmatics and risk**: The management aspects of the project. These aspects consider the cost, schedule, and risk of building, fielding, and supporting the solution to include the cost, schedule, and risk for changing the necessary business processes.

The interface architecture defines the interface specification based on all the system internal and external interface requirements. A baseline interface definition must be agreed upon before the beginning of implementation activities, and the user interface must be defined and maintained as an integral part of the system specification. Interface control and management is an important part of the integration architecture that results in the management of communication, coordination, and responsibility across a common boundary between two organizations, phases, or physical entities which are interdependent. It ensures that internal and external interfaces are properly identified, integrated, stabilized, and controlled early in order to prevent expensive and time-consuming fixes later.

Semantic integration addresses the semantic content of data in different systems. It is more important to be aware of inconsistencies in the semantics rather than to eliminate the differences, which may be difficult, especially in systems from different vendors. Semantic consistency refers to the concern for consistency at the data level. Once the information system components and equipment are interconnected and operational, users are able to access systems and manipulate data (create, retrieve, modify, and delete) across various operational environments. For this reason, the implementation of semantic integration is essential to prevent data duplication, redundancy, and instability [Mische, 1998].

The qualification architecture involves defining and managing the test activities as shown in Figure 5. Test approaches are developed to provide the objectives, schedule, environment requirements, and entry and exit criteria for the test stage. Based on these approaches tests are planned and prepared to identify test conditions, and test cycles, and to define the input data and expected results. It is important to establish the appropriate test environments and to ensure that they are tested prior to the execution. Test cases and sequence for test executions are derived based on the system requirements and the test approaches.

## 5.3. Integration Planning and Implementation Subprocess

Figure 6 shows all the activities in the integration planning and implementation subprocesses. The integration require-
ments and architecture are analyzed to identify the risks associated with systems integration. The risk analysis occurs concurrently as the requirements and architectures evolve. All the factors that impact integration and testability of the system are identified. These risk factors and their mitigation activities need to be included in the integration plan. An integration plan is based on the integration specification elicited from the requirements and architecture. The integration plan includes detailed integration implementation, verification, and validation activities, execution sequences, dependencies, resources, tools, executions states and modes, assessment of integration, and management of these activities. During this subprocess traceability across all integration process related activities are verified. The integration plan and all specifications are documented. Configuration management of the plan and the specifications plays a critical part in the success of integration. Verification and validation of the integration plan is performed by testing the system prototype. In most of the system development efforts, the proposed system is modeled and optimized by developing prototypes of the system. These prototypes are based on the derived requirements and architecture. They are used to understand the behavior of the
system in its development and operational environment. This helps in identifying and mitigating the risks at an early stage. Once the subsystems and components are built or acquired, the integration activities are implemented as per the integration plan. Some other subactivities that will be involved in this Integration Planning and Implementation Subprocess will be: monitor status of components and subsystems at lower levels of procurement or integration, receive components and subsystems, check conformance of components and subsystems to their specifications, prepare integration environment, develop integration tools and facilities, and develop integration procedures and certify integration personnel.

5.4. Integration Validation and Verification Subprocess

Figure 7 shows the activities of the verification and validation (V&V) subprocess. Verification and validation of the subsystems and components that are built or acquired follows the qualification plan developed in the earlier phase. Verification establishes the truth of correspondence between a product/system and its specification. The activities in this V&V subprocess verify and validate if we are building the right product/system and at the same time building it right. Validation is the act of ensuring that the system works as per its intended functionality and that the users are satisfied and willing to accept the system. An example is the comparison of the actual system response to an online transaction to what was originally expected, requested, and finally approved for an online transaction processing system. Validation establishes the suitability of a product/system for its operational mission in a given environment based on operational or field testing. Both verification and validation activities should be traced back to the system requirements.

During this subprocess the test scripts are verified and validated, test environment is developed and the tests are executed in its sequence. These activities are based on the developed qualification plan. There are two types of testing, Functional/Life Cycle Approach and Structural. In functional testing, test the functionality of the system and ensure that the user functional requirements and specifications are met. Test conditions are generated to evaluate the correctness of the application. These include unit test, assembly test, systems integration test, operational readiness test, and user test. In structural testing, we test the structural and physical capability of the system and ensure that the system is structurally and technically sound, can perform the intended tasks, and that the components integration works cohesively. These include fatigue testing, strain and impact testing, branch and path testing, control flow testing, data flow testing, security testing, and stress/volume/scalability testing. The test results are logged and documented. The errors are fixed through rework and change control management.

The SIPM process model has been primarily proposed to facilitate baselining of as-is SI processes in an organization, benchmarking them against targeted optimal levels, and reengineering them for better integration effectiveness as a result of controlling complexity of SI. The model would help or-

Figure 7. Integration implementation, integration verification, and validation activities flowchart. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]
organizations identify areas of SI that are strong and others that require more attention. It would also highlight critical attributes of SI in their specific organizational context.

6. APPLICATION OF THE SYSTEMS INTEGRATION PROCESS MODEL (SIPM) FOR PROCESS ANALYSIS AND IMPROVEMENT

This section describes our experience of applying the SIPM to analyze, evaluate, optimize, and recommend improvements to the current systems integration process and activities of a government agency (as observed and reported on nine different development projects). We will be referring to this government agency as the “Agency” in the rest of this paper.

A critical component of undertaking process analysis and redesign is to identify the gaps and redundancies and implement improvements in order to make processes efficient, effective, and adaptable [Harrington, 1995]. This paper illustrates our findings of using the SIPM as a baseline to analyze and improve the current SI process of the Agency. The next section will cover our methodology for conducting this work for the Agency.

7. RESEARCH METHODOLOGY

This section provides an overview of the research methodology used for the pilot application of SIPM for SI process reengineering. The research sample was nine system development projects in the agency. The project data were provided by the Systems Engineering Leads of each of the nine projects on the scope of the SI process and related activities that were being conducted.

We used the survey research method to identify and analyze the Agency’s current Systems Integration process and activities. The survey includes 45 questions based on the activities of SIPM. Each question is asked to determine if the specific activity is currently performed on each of the nine projects. Respondents were asked to select one of the three given answers of “Yes”; “Not exactly but we perform similar activities”; or “No”. A free-form comments section was also provided at the end of the survey. The survey responses were consolidated, reviewed, and analyzed.

The as-is analysis of the Agency’s current SI process was based on five focus areas (research questions): (1) gap analysis between the Agency’s current SI process and SIPM; (2) strengths and weaknesses of the Agency’s current SI process; (3) effectiveness of the integration requirements subprocess in terms of completeness, refinement, and traceability; (4) effectiveness of the Agency’s current SI process to address critical SI issues such as COTS, legacy, interoperability, configuration management, interface control and management, and adherence to standards and regulations; and (5) quality of the integration verification and validation. The survey results were analyzed based on these research questions. The research questions, survey results, and the analysis are discussed in detail in the following section.

8. RESEARCH FINDINGS

The survey results and their analysis not only demonstrated to us the level of effectiveness of the Agency’s current SI process but also revealed the areas which need improvement. Based on these findings, a set of recommendations was provided to the Agency to improve its current SI process and activities that require attention. This section is organized by research questions and their relevant analysis and findings.

The results are shown as percentage values which indicate the relative difference between the proposed SIPM and Agency’s current process and activities. This difference is calculated by assigning a weight of 2 to “Yes” answers which were given for a specific activity, a weight of 1 to “Not exactly, but we perform similar activities’ answers,” and assigning a weight of 0 to “No” answers. A weighted sum of the answers were calculated for a given a specific activity. We then identified the gaps based on an assumption that if all projects reported a “Yes” for an activity then that activity will be considered as having a 0% gap or, in other words, the activity satisfies requirements of out SIPM completely (at 100%).

8.1. Gap Analysis of the Agency’s Current SI Process and SIPM

Based on the survey results, we found that the overall gap between the Agency’s current SI process and the proposed SIPM is 28% (Fig. 8). This indicates that the Agency’s current SI process is not comprehensive enough (28% deficient) in scope to include all aspects of an end-to-end systems integration approach. A subprocess by subprocess gap analysis reflected that the Integration Architecture Subprocess in particular requires more reengineering. This subprocess has a 40% variance with respect to the integration architecture subprocess in SIPM. Addressing integration issues and design decisions earlier in the development lifecycle will reduce the error and fault rate and as well as rate of rework. This will in turn provide programmatic advantage in terms of cost, schedule and resources. [Jain et al., 2008] shows that the system architecture and requirements impact the systems integration complexity and quality of verification and validation. The integration architecture subprocess is highly dependent on the system architecture and requirements subprocesses. The tradeoff decisions critical to systems integration are addressed during this subprocess of SI. To improve the effectiveness of the SI process, the Agency needs to provide more emphasis on the integration architecture activities.

The gap analysis also shows that the Agency’s current SI process was relatively stronger in the Integration Implementation sub-process and Integration Verification and Validation sub-process activities (variance with respect to the corresponding sub-process in SIPM less than 15%). This shows that the Agency’s current SI practices focus mainly on the integration implementation, verification and validation. In other words, the Agency’s significant focus is on the downstream activities of systems integration. This finding resonates with our early discussion on how the current SE practices consider systems integration as the physical integration of subsystems and components and their verification and validation. However, SIPM takes a different stand and suggests a comprehen-
sive end-to-end approach which requires emphasis on integration starting from the earlier front-end sub-processes of SE lifecycle.

8.2. Strengths and Weaknesses of the Agency’s Current SI Process

We analyzed the survey results in order to identify the strengths and weaknesses of the Agency’s current SI process. This was done to better understand the current state of systems integration at the Agency as reported by the nine projects. Identification of the weak areas within the Agency’s current SI process helped us to focus on the areas which required more attention and which needed to be improved in order to achieve a more effective integration process.

We identified the different SI activities in the Agency and their corresponding gaps. The size of the gap is indicative of the strengths and weaknesses of the Agency’s SI process. The bigger the gap the weaker is the Agency on those SI activities and vice versa.

Figure 9 shows strengths and weaknesses of the Agency’s current SI process. The overall SI process gap of 28% is shown by the horizontal line. The x-axis shows the SI process subprocesses and the y-axis shows the level of gap. The higher the vertical placement of the bubble above the 28% line, the bigger is the gap. The size of the bubbles indicates the number of activities which fall within the level of difference on the y-axis. For example, if we look at the lower left quadrant of the figure, where the x-axis shows “Integration Requirements Activities,” we see a bubble marked with “2” at the 11% level on the y-axis. This bubble indicates that currently there are 2 activities within the Integration Requirements Activities which have 11% gap from the SIPM level.

Our focus in this research was to recommend improvements for optimizing the current state of Agency’s SI process and related activities. Understanding the Agency’s strengths helped us to develop and recommend a process model which would be suitable with its given strengths. We also identified and analyzed the deficiencies of the Agency’s SI process and pointed out the risks arising out of these deficiencies. As a result, the areas which were identified as weaknesses of Agency’s current SI process were rated as highest priority for improvement. Our approach was to develop a model for the Agency that eliminated the inefficiencies resulting from the weak areas of the Agency’s current SI process. This would help contain the risks resulting from such SI process inefficiencies.

8.3. Integration Requirements Issues

As discussed earlier, integration requirements subprocess is one of the most important subprocesses within SI process. It provides the front-end basis for effective systems integration. We analyzed the survey results to identify integration requirements issues within Agency’s current SI process. We looked at two main issues within integration requirements: (i) emphasis on different types of integration requirements and (ii) refinement of different types of requirements. We found that the Agency is relatively good in identifying stakeholders and gathering and analyzing the systems integration requirements. Overall 78% of the requirements integration related activities are currently being conducted at the Agency.

8.3.1. Emphasis on Different Types of Requirements

We analyzed the survey results to identify the level of emphasis (or lack thereof) given to the different types of requirements. Figure 10 shows the level of emphasis given to different types of requirements. The three areas on which the agency focuses were qualification, standards, and COTS requirements.

Interface, integration technology, interoperability, and legacy requirements received the least emphasis. This was indicative of a substantial potential for improvement. Identification, analysis, and management of interoperability, legacy, integration technology, and interface requirements are extremely critical in defining, constraining, and facilitating systems integration. The lack of emphasis on these types of
Figure 9. Strengths and weaknesses of the Agency’s current activities. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Figure 10. Level of emphasis on different types of requirements. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]
requirements can lead to integration issues and jeopardize the effectiveness and efficiency of systems integration process.

### 8.3.2. Refinement of Originating Requirements

Our SI\(^\text{PM}\) goes beyond the gathering of originating integration requirements and emphasizes on their refinement as well. We found that the Agency’s current SI process included a fairly strong requirements refinement process. The deriving requirements process received only 4\% less emphasis than the collecting originating requirements (Fig. 11). This finding might indicate that some of the requirements categories were not identified in the beginning or not gathered from stakeholders. As a result they had to be derived at the later stages. Another reason for this difference might reflect that some types of originating requirements could not be collected completely and therefore had to be derived at the later stages. We assume that a significant portion of this gap is contributed by the missing design-tradeoff requirements primarily related to COTS and legacy integration.

### 8.4. The Role of Systems Integration Issues Impacting Agency’s Current SI Process

Some of the common systems integration related issues are interoperability, COTS and Legacy Integration, Interface Control and Management, Standards, Guidelines and Recommendations, and Configuration Management. This section discusses how effectively the Agency’s current SI process addresses these issues. Our findings are summarized in Figure 12. Recommendations on adoption of these integration issues related SI\(^\text{PM}\) activities were provided to the Agency for optimization and reengineering.

#### 8.4.1. Interoperability Issues

Interoperability issues directly impact the success of systems integration process. The 38\% variance (Fig. 12) between the Agency’s current interoperability related activities and SI\(^\text{PM}\) shows a lack of focus on interoperability issues. Interoperability of a system is addressed through the interfaces, compliance, and design tradeoffs. SI\(^\text{PM}\) proposes early identification of interoperability requirements. These requirements help in identifying the respective compliance requirements and other design constraints at an early stage of development. Interfaces are also directly impacted by these requirements. SI\(^\text{PM}\) also focuses on traceability of these requirements from their origin to their implementation, verification and validation.

#### 8.4.2. COTS and Legacy Integration Issues

In today’s world, the common challenges faced when engineering a system are related to the integration and interoperability with COTS systems and existing legacy systems. The gap analysis (Fig. 12) shows that the Agency’s focus on the COTS and legacy integration issues are not adequate. Adoption of COTS and legacy systems are driven by the time-to-

Figure 11. Gap between SI\(^\text{PM}\) and As-Is Requirements Subprocess. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Figure 12. Issues impacting Agency’s current systems integration process. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]
market constraints and the higher level of technology readiness of the COTS products. But the longevity and lack of design flexibility with COTS and legacy systems constrain the system integration. Upfront COTS and legacy integration decisions can help in identifying the appropriate interface and performing the adequate V&V. SIPM supports early COTS and legacy system integration.

8.4.3. Role of Interface Control and Management
The gap analysis (Fig. 12.) shows that the Agency’s focus on interface control and management has lesser variance compared to other integration issues. Good interfaces enable better system integration and interoperability. The SIPM includes activities which foster adaptable and scalable interface control and management. Interface control and management ensures that internal and external interfaces are properly identified, integrated, stabilized, and controlled early-on in order to prevent expensive and time-consuming fixes later.

8.4.4. Role of Standards, Recommendation, and Guidelines in the SI Process
The usage of standards, recommendations and guidelines is important for SI process to ensure interoperability, to minimize interface issues and to streamline development efforts. According to survey results, standards, recommendations, and guidelines play an important role in the Agency’s current SI process. This reflected the regulatory nature of development that was unique to the Agency. Though the variance was only 22% (Fig. 12) between the Agency’s current state and the SIPM, we still recommended the Agency focus more on them as the consequences of not including such constraining requirements in the SI process could be severe. SIPM emphasizes the use of standards, recommendations, and guidelines to achieve uniformity and consistency in design thereby leading to simple systems integration.

8.4.5. Role of Configuration Management in SI Process
Configuration Management (CM) is another important aspect of SIPM. The gap analysis (Fig. 12) shows a 26% variance between the Agency’s configuration management and the corresponding SIPM activities. SIPM includes activities for managing the configuration of requirements, specification, and test results. Weak configuration management activities make it difficult to control and manage not just the systems integration process but also the other development subprocesses. The variance in configuration management is a result of the divulged focus on configuration due to very high focus on documentation in the Agency’s development environment. The Agency was recommended to provide more focus on change controls and thereby improve its configuration management.

8.5. Quality of Integration Verification and Validation
The quality of integration verification and validation (V&V) directly impacts the overall outcome of SI Process. In the SIPM the prerequisites of the integration verification and validation subprocess are addressed in the upstream activities of SI process. These upstream activities provide the required artifacts and requirements for an upfront and effective planning of the V&V sub-process. Ten such supporting activities of V&V from other subprocesses of SIPM were identified: (i) Integration Requirements Subprocess: (Derive Qualification Requirements, Derive Test Cases); (ii) Integration Architecture Subprocess: (Develop Semantics for Integration, Develop Semantic Specification, Develop Qualification Architecture, Develop Test Sequence, Develop Test Environments Specifications, Test Architecture); and (iii) Integration Planning Subprocess: (Develop System Prototype, Test System Prototype). These 10 activities along with the five V&V activities aid in achieving better quality of integration verification and validation.

Figure 13 shows the minimum, maximum, and average gap values for these activities between our SIPM stipulations and those that were reported by the Agency’s projects for each of these activities. This analysis shows us that the downstream activities that result in better quality of V&V are provided more emphasis than the upstream activities. The average gap values were less than 15% in the Integration Planning and Integration Verification and Validation subprocesses. This analysis also shows that more importance needs to be given to the Integration Architecture subprocess (higher average gap value of 45%) and thereby improve the quality of V&V.

9. CONCLUSIONS AND FUTURE WORK
The Systems Integration Process Model (SIPM) is developed and applied to analyze one organization’s current Systems Engineering DOI 10.1002/sys
Integration process and activities, and recommend improvements. Most available standards, maturity models, and methodologies address systems integration issues partially and do not take a holistic end-to-end approach to systems integration. Due to the emerging challenges, the need for a holistic approach to systems integration has become critical. Our Systems Integration Process Model is developed as a result of extensive research in the field and intends to provide a comprehensive approach to systems integration. The application of our SIPM to evaluate, analyze, and optimize one organization’s current systems integration process and activities provided us an opportunity to implement and test our Model. The Model is in its early phases of being piloted. There are at least two doctoral-level researchers who are working on further testing and refining it. Our hope is that in the future applications we will be able to define metrics related to each of the 45 SI activities. Our future work will focus on defining different levels of SI process maturity based on the metrics for each of the SI activities. The maturity level of the SI process will indicate the likelihood for success or failure of the product or project.

REFERENCES

C. Beise, A model of the is/organizational interface and users’ perceptions of is effectiveness, Computur Personnel (July 1994).
R. Jain, A. Chandrasekaran, G. Elia, and R. Cloutier, Exploring the impact of systems architecture and systems requirements on systems integration complexity, IEEE Syst J (June 2008).

Rashmi Jain is an Associate Professor of Systems Engineering at Stevens Institute of Technology. Dr. Jain has over 15 years of experience of working on Information Technology (IT) systems. Prior to joining Stevens she was with Accenture (formerly known as Andersen Consulting). Over the course of her career she has been involved in leading the implementation of large and complex systems engineering and integration projects. She has given invited lectures internationally at Keio University, and Shibaura Institute of Technology, Japan, Overseas Chinese Institute of Technology (OCIT), Taiwan, Indian Institute of Technology—Delhi etc. She is a visiting professor for System Architecture and Integration at Keio University. Her teaching and research interests include systems integration, systems architecture and design, business process reengineering, and rapid systems engineering. Dr. Jain has authored several papers on these topics. She holds Ph.D. and M.S. degrees in Technology Management from Stevens Institute of Technology.

Anithashree Chandrasekaran is a Doctoral Candidate in the School of Systems and Enterprises at Stevens Institute of Technology. She is also working as a technology risk manager. Her research interests include rapid systems development and its processes, development process reengineering, risk management and modeling, system integration, system design and architecture. She is currently working on developing a dynamic risk assessment model for rapid system development process. She obtained her B.E. in Electrical and Electronics Engineering from P.S.G. College of Technology, India. She obtained her M.S. in Systems Engineering from Stevens Institute of Technology. She served as president and also as a founding member of the Stevens INCOSE student chapter.
Ozgur Erol is a Ph.D. student in Systems Engineering and Engineering Management at Stevens Institute of Technology. Her research interests include systems integration process evaluation, management, and improvement. She received her bachelors and masters degrees in Industrial Engineering from Istanbul Technical University, Turkey and her MBA degree from Saint Joseph’s University, Philadelphia, PA. She has worked in several information technology and organizational reengineering projects prior to joining the Ph.D. program at Stevens.

<enote>AQ1: Please list in References
<enote>AQ2: Please give volume number and page range
<enote>AQ3: Please give page range