2022-09

EXPLOITING KASPAROV'S LAW: ENHANCED INFORMATION SYSTEMS INTEGRATION IN DOD SIMULATION-BASED TRAINING ENVIRONMENTS

Morse, Matthew M.

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/71082

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

EXPLOITING KASPAROV'S LAW: ENHANCED INFORMATION SYSTEMS INTEGRATION IN DOD SIMULATION-BASED TRAINING ENVIRONMENTS

by

Matthew McCamley Morse

September 2022

Dissertation Supervisor: Curtis L. Blais

Research for this dissertation was performed at the MOVES Institute.

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2022		**3. REPORT TYPE AND DATES COVERED** Dissertation	
4. TITLE AND SUBTITLE EXPLOITING KASPAROV'S LAW: ENHANCED INFORMATION SYSTEMS INTEGRATION IN DOD SIMULATION-BASED TRAINING ENVIRONMENTS				**5. FUNDING NUMBERS**
6. AUTHOR(S) Matthew McCamley Morse				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A				**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.				**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Despite recent advances in the representation of logistics considerations in DOD staff training and wargaming simulations, logistics information systems (IS) remain underrepresented. Unlike many command and control (C2) systems, which can be integrated with simulations through common protocols (e.g., OTH-Gold), many logistics ISs require manpower-intensive human-in-the-loop (HitL) processes for simulation-IS (sim-IS) integration. Where automated sim-IS integration has been achieved, it often does not simulate important sociotechnical system (STS) dynamics, such as information latency and human error, presenting decision-makers with an unrealistic representation of logistics C2 capabilities in context. This research seeks to overcome the limitations of conventional sim-IS interoperability approaches by developing and validating a new approach for sim-IS information exchange through robotic process automation (RPA). RPA software supports the automation of IS information exchange through ISs' existing graphical user interfaces. This "outside-in" approach to IS integration mitigates the need for engineering changes in ISs (or simulations) for automated information exchange. In addition to validating the potential for an RPA-based approach to sim-IS integration, this research presents recommendations for a Distributed Simulation Engineering and Execution Process (DSEEP) overlay to guide the engineering and execution of sim-IS environments.

14. SUBJECT TERMS			**15. NUMBER OF PAGES**
distributed simulation, staff training exercise, RPA, sim-C2, sim-IS, LVC, simulation interoperability, logistics, logistics information system			411
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

EXPLOITING KASPAROV'S LAW: ENHANCED INFORMATION SYSTEMS INTEGRATION IN DOD SIMULATION-BASED TRAINING ENVIRONMENTS

Matthew McCamley Morse
Major, United States Marine Corps
BS, United States Naval Academy, 2008
MS, Boston University, 2011
MS, Naval Postgraduate School, 2014

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
September 2022

Approved by: Imre L. Balogh
Department of
MOVES Institute
Dissertation Chair

Curtis L. Blais
Department of
MOVES Institute
Dissertation Supervisor

Rudolph P. Darken
Department of
Computer Science

Douglas J. MacKinnon
Department of
Information Sciences

Perry L. McDowell
Department of
MOVES Institute

Katherine L. Morse, JHU/APL

Approved by: Gurminder Singh
Chair, Department of Computer Science

Joseph P. Hooper
Vice Provost of Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Despite recent advances in the representation of logistics considerations in DOD staff training and wargaming simulations, logistics information systems (IS) remain underrepresented. Unlike many command and control (C2) systems, which can be integrated with simulations through common protocols (e.g., OTH-Gold), many logistics ISs require manpower-intensive human-in-the-loop (HitL) processes for simulation-IS (sim-IS) integration. Where automated sim-IS integration has been achieved, it often does not simulate important sociotechnical system (STS) dynamics, such as information latency and human error, presenting decision-makers with an unrealistic representation of logistics C2 capabilities in context. This research seeks to overcome the limitations of conventional sim-IS interoperability approaches by developing and validating a new approach for sim-IS information exchange through robotic process automation (RPA). RPA software supports the automation of IS information exchange through ISs' existing graphical user interfaces. This "outside-in" approach to IS integration mitigates the need for engineering changes in ISs (or simulations) for automated information exchange. In addition to validating the potential for an RPA-based approach to sim-IS integration, this research presents recommendations for a Distributed Simulation Engineering and Execution Process (DSEEP) overlay to guide the engineering and execution of sim-IS environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A2C2	Adaptive Architectures in Command and Control
AI	artificial intelligence
ALCES	Aggregate Level Communications Effects Server
AUI	abstract user interface
BCS3	Battle Command Sustainment Support System
BFG	Batch File Generator
BFT	Blue Force Tracker
BITA	business process-information technology alignment
BML	Battle Management Language
BP	best practices
BP-IS	business process-information system
BPEL	business process execution language
BPM	business process model
BPMN	business process modeling notation
BPMS	business process management system
BPR	business process redesign
BSC	battle simulation center
BSTX	battle staff training exercise
C2	command and control
C2PC	C2 personal computer
C2SIM	C2 to Simulation Interoperability
CASCOM	Combined Arms Support Command
C-BML	Coalition-BML
CCIR	commander’s critical information requirement
CDL	choreography description language
CLB	combat logistics battalion
CLC2S	Common Logistics C2 System
CLR	combat logistics regiment
CM	conceptual model
CMMS	Conceptual Model of the Mission Space

COC	combat operations center
CoP	community of practice
CPX	command post exercise
CRF	Chameleon Reference Framework
CSE	cognitive systems engineering
CSO	Core Software Ontology
CTA	cognitive task analysis
CTT	concur task trees
CUI	concrete user interface
CWA	cognitive work analysis
DCIPS	Defense Casualty Information Processing System
DCT	Decision Centered Testing
DDOS	distributed denial of service
DiaMODL	dialogue modeling language
DIS	Distributed Interactive Simulation
DLA	Defense Logistics Agency
DMAO	DSEEP Multi-Architecture Overlay
DMSC	Dynamic Model of Situated Cognition
DMSO	DOD Modeling and Simulation Office
DO	domain ontology
DOD	Department of Defense
DoDAF	DOD Architecture Framework
DSEEP	Distributed Simulation Engineering and Execution Process
DSL	domain specific language
DSR	design science research
DSS	decision support system
EDIPI	electronic data interchange personal identifier
EID	ecological interface design
EPC	event-driven process chain
ERP	enterprise resource planning
ERPSim	ERP Simulation
FEAT	Federation Engineering Agreements Template

FlowiXML	user interfaces to workflow based on UsiXML
FOM	federation object model
FUI	final user interface
GCSS-MC	Global Combat Support System – Marine Corps
GOMS	goals, operators, methods, and selection
GPS	Global Positioning System
GUI	graphical user interface
HAGA	humans-are-good-at
HIMARS	high mobility artillery rocket system
HitL	human-in-the-loop
HLA	High Level Architecture
HMMWV	high mobility multipurpose wheeled vehicle
HTA	hierarchical task analysis
IAS	Intelligent Automation Services
ICODES	Integrated Computerized Deployment System
IDEF	integrated computer-aided manufacturing definition
IEEE	Institute of Electrical and Electronics Engineers
IRB	institutional review board
IS	information system
ISML	interface specification meta-language
IT	information technology
ITV	in-transit visibility
IVIS	in-vehicle information systems
JAIC	Joint Artificial Intelligence Center
JCATS	Joint Conflict and Tactical Simulation
JCS	joint cognitive system
JDLM	Joint Deployment Logistics Model
JLVC	Joint Live Virtual Constructive
JNE	Joint Network Emulator
JTDS	Joint Training Data Services
LESD	Logistics Exercise and Simulation Directorate
LogAIS	logistics automated information system

LSE	large scale exercise
LVC	live, virtual, constructive
LVC-G	live, virtual, constructive and gaming
M&S	modeling and simulation
MAGA	machines-are-good-at
MAGTF	Marine Corps Air Ground Task Force
MARIA	model-based language for interactive applications
MBUID	model-based user interface development
MCAGCC	Marine Corps Air Ground Combat Center
MCLOG	Marine Corps Logistics Operations Group
MCO	modeling concepts ontologies
MIT	Massachusetts Institute of Technology
MMP	macrocognitive modeling procedure
MOSA	modular open systems approach
MOVES	Modeling, Virtual Environments, and Simulation
MRE	meal ready to eat
MSDL	Military Scenario Definition Language
MSEL	master scenario events list
MSG	Modeling and Simulation Group
MSTP	MAGTF Staff Training Program
MTVR	medium tactical vehicle replacement
MTWS	MAGTF Tactical Warfare Simulation
MVCM	Multi-Viewpoint Conceptual Modeling
N2ES	Network Effects Emulation System
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NDM	naturalistic decision-making
NPS	Naval Postgraduate School
NSN	national stock number
NSSC	NASA Shared Services Center
OBS	Order of Battle Service
ONR	Office of Naval Research

OOTLUF	out-of-the-loop-unfamiliarity
OTC	observer/trainer/controller
OTH	over-the-horizon
OUSD-C	Office of the Under Secretary of Defense – Comptroller
PA	process alignment
PRNG	pseudo random number generator
RFID	radio-frequency identification
RNG	random number generator
RPA	robotic process automation
RPD	recognition-primed decision-making
RPR	real-time platform reference
SAP	Systems, Applications, Products in Data Processing
SBT	scenario-based training
SCOR	supply chain operations reference
SDEM	simulation data exchange model
SDT	system detection theory
sim-C2	simulation-C2
sim-IS	simulation-information system
SINEX	Scenario Initialization and Execution
SISO	Simulation Interoperability Standards Organization
SoaML	service oriented architecture modeling language
SSDM	Sea Service Deployment Module
STRATIS	Storage Retrieval Automated Tracking Integrated System
STS	sociotechnical system
SysML	Systems Modeling Language
TADMUS	Tactical Decision-Making Under Stress
TAMCN	table of authorized material control number
TENA	Test and Training Enabling Architecture
TERESA	transformation environment for interactive systems representations
TCPT	Transportation Capacity Planning Tool
TSO	Technology Services Organization
UI	user interface

UIDL	user interface description language
UIML	user interface markup language
UMCO	unified modeling concepts ontology
UML	Unified Modeling Language
U.S.	United States
UsiXML	user interface extensible markup language
VBS	Virtual Battlespace
VV&A	verification, validation, and accreditation
WFMS	workflow management system
WSMM	work system modeling method
WST	work system theory
XIML	extensible interface markup language
XUL	extensible modeling language user interface language

EXECUTIVE SUMMARY

The proliferation of increasingly powerful information systems and associated business processes in the modern workplace and on the battlefield necessitates staff training environments that represent the information systems (IS) and associated sociotechnical system (STS) dynamics. Such environments are needed to exercise the organizational processes and information systems and to develop the competencies of personnel in context. Despite advances in the representation of some command and control (C2) systems in simulation-supported staff training environments, limitations remain in the representation of human-in-the-loop (HitL) information systems, including many logistics information systems, and the STS dynamics which influence how these information systems are populated in the operating environment. Existing means of integrating simulations and C2 systems are insufficient for supporting automated simulation-information system (sim-IS) information exchange and simulation of STS dynamics.

This research explored how robotic process automation (RPA) can facilitate a new approach for automating sim-IS information exchange and the simulation of STS dynamics. A design science research (DSR) methodology was employed to determine whether an RPA-based approach to sim-IS information exchange can support automated sim-IS information exchange and the simulation of STS dynamics, and how such RPA-based sim-IS environments can be designed and developed. These research questions were addressed through the development and validation of two DSR artifacts: an instantiation artifact in the form of a prototype RPA-based architecture for sim-IS information exchange and a method artifact in the form of recommendations for a sim-IS environment overlay for the Distributed Simulation Engineering and Execution Process (DSEEP) (*IEEE Std 1730*, 2011). The instantiation artifact was validated, in the DSR sense of the word, for its ability to support automated sim-IS information exchange and simulation of specified STS dynamics. The STS dynamics simulated include temporal dynamics (information latency and timeliness variability) and information content degradation (accuracy, precision, and completeness variability). This was achieved through a modeling and simulation (M&S)

verification and validation process, including quantitative and qualitative analysis of the instantiation artifact in laboratory and field environments.

Verification of the RPA-based sim-IS architecture for its capacity to support automated sim-IS information exchange and simulation of specified STS dynamics was conducted with quantitative analysis of the prototype's performance in two sim-IS environments within a controlled laboratory environment. In one sim-IS environment an aggregate constructive simulation, the Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS), was integrated with the Marine Corps' Common Logistics Command and Control System (CLC2S). In the other sim-IS environment, an entity-level constructive simulation, the Joint Conflict and Tactical Simulation (JCATS), was integrated with CLC2S. In both sim-IS environments the constructive simulation and HitL logistics information system were integrated using the RPA-based sim-IS information exchange architecture seen in Figure 1.

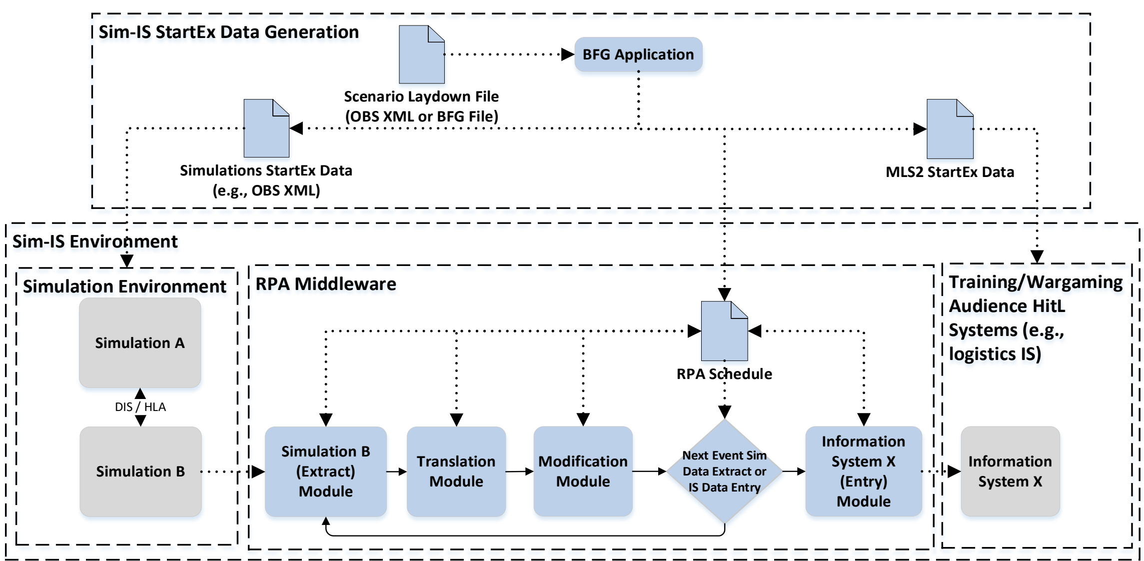


Figure 1. RPA-based sim-IS information exchange architecture

Results from the quantitative analysis of prototype architecture's performance in the two sim-IS environments indicated that an RPA-based approach to sim-IS information exchange can support automated sim-IS information exchange and the simulation of STS dynamics. For the simulation of temporal dynamics, simulation of both timeliness and latency distributions was found to adequately align with the target distributions, with chi square goodness of fit test values for each scenario exceeding the 0.95 threshold. The simulation of information content degradation was observed with practical but not statistical significance. While the architecture was found to provide adequate accuracy and precision for sim-IS information exchange, determining the timeliness of RPA-based sim-IS information exchange was found to require consideration of the particular sim-IS environment(s) to be supported. A technique for addressing this issue in the design and development of RPA-based sim-IS environments was developed and addressed in the recommendations for a DSEEP overlay for sim-IS environments.

Results from the quantitative analysis were necessary but not sufficient to support validation of the RPA-based sim-IS architecture. A demonstration of the architecture for domain subject matter experts (SMEs) in a field environment was conducted to support qualitative validation of the artifact's utility for supporting its intended use. The results of the demonstration and subsequent SME interviews indicated that the proposed RPA-based architecture would support the intended use in facilitating sim-IS environments for staff training. This includes the potential for supporting the representation of additional HitL information systems in staff training environments and the simulation of STS dynamics otherwise too manpower or cost prohibitive to represent.

The method artifact, recommendations for a sim-IS overlay for the DSEEP, was developed to facilitate the design and development of sim-IS environments which represent target integrated business processes and associated STS dynamics, including use of an RPA-based approach to sim-IS information exchange. While the DSEEP provides guidance for the engineering and execution of distributed simulation environments, it does not provide guidance necessary for supporting sim-IS environments. This research presents 43 specific recommendations for issues to be addressed in a DSEEP overlay to guide the engineering and execution of sim-IS environments, with an emphasis on supporting RPA-based sim-IS information exchange. These recommendations were developed concurrently with the design, development, and validation of the RPA-based sim-IS architecture. Many of the recommendations provided are directly associated with obstacles identified in the design, development, and testing of the prototype RPA-based sim-IS architecture and the tools and techniques developed to overcome those obstacles.

In addition to presenting and validating a new, RPA-based approach to sim-IS information exchange and simulation of STS dynamics for staff training, this research presents new opportunities for the design of sim-IS environments in support of other domains. The low overhead, modular sim-IS environments facilitated by an RPA-based approach to sim-IS information exchange may support a new means of wargaming environments with enhanced representation of real-world HitL information systems and the integrated business processes through which they are populated. It may also facilitate a flexible environment for the continuous coevolution of integrated business processes themselves. This research presents a first step in the exploration and development of an RPA-based approach to sim-IS information exchange, one which can support increased representation of HitL information systems in simulation-supported environments in support of staff training and other important problem spaces.

References

IEEE Std 1730–2010. (2011). IEEE recommended practice for distributed simulation engineering and execution process (DSEEP). IEEE Computer Society.

ACKNOWLEDGMENTS

The completion of this dissertation was possible only thanks to the immense support and guidance generously provided to me by many mentors, friends, and family. I would first like to thank my advisor, Dr. Curtis Blais, for the immense time and effort he devoted to this work. I could not have hoped for a more passionate or diligent mentor to guide me through this process. I would also like to thank my exceptional dissertation committee members: Dr. Imre Balogh, Dr. Katherine Morse, Dr. Douglas MacKinnon, Dr. Rudy Darken, and Perry McDowell. Special thanks to Dr. Morse (no relation) for our many discussions regarding simulation conceptual modeling and what the world would look like if everyone would just follow the DSEEP and to Dr. Balogh for his steadfast leadership of the MOVES Institute within which I've had the great pleasure of working for these three years. My thanks to Dr. Darken for many insightful, powerful questions and to Dr. MacKinnon for selflessly providing guidance on this process long before I even began it. Thank you to my fellow doctoral students, namely LtCol Charlie Rowan, Col Joe Novak, and LtCol Scotty Black, for our regular oral board sparring sessions and daily exchanges over these three years. To my parents, thank you for cultivating a love of learning and a love of hard work. Finally, I would like to thank my wife Melisa, for her endless support and for accompanying me on this journey and the many to come.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Shortly after his 1997 defeat at the "hands" of IBM's Deep Blue, chess grandmaster Garry Kasparov began exploring the role of computers in support of human decision-making. He hosted and observed chess competitions which pitted human-machine chess teams against one another. It was in these "advanced chess" competitions that Kasparov first noted that a "weak human + machine + better process was superior to a strong machine alone and, more remarkably, superior to a strong human + machine + inferior process" (Kasparov, 2017, p. 246). This idea that the process is more powerful than either the human or the information system alone, referred to by some as "Kasparov's Law" (Kasparov, 2017, p. 247), has important implications for the design of modern staff training environments.

While Kasparov was focused on the use of decision support systems leveraging artificial intelligence, Kasparov's Law also applies to more conventional information systems. As increasingly powerful and complex computerized information systems (ISs) proliferate in work environments, user competencies for leveraging ISs must be developed and evaluated relative to the broader integrated business processes. For military and emergency management environments, where real-world opportunities for exercising processes are limited, the development of IS-supported decision-making requires scenario-based training environments. This is particularly true for naturalistic decision-making (NDM) environments, time-constrained decision-making environments characterized by uncertainty and high risk.

Existing simulation-supported scenario-based training environments are limited in their ability to stimulate human-in-the-loop (HitL) ISs or simulate the sociotechnical system (STS) dynamics (e.g., information latency, information entry errors) which influence the effectiveness of integrated business processes and the development of decision-makers' intuition. Conventional mechanisms for integrating simulations and ISs are limited in their ability to support the representation of the myriad HitL ISs and STS dynamics seen in operational environments. This research explores a new means of integrating simulations and HitL ISs, through robotic process automation (RPA), to support the inclusion of more HitL ISs in simulation-supported, scenario-based training environments and the simulation of STS dynamics. Before discussing the limitations of conventional simulation-IS (sim-IS) interoperability approaches, it is important to first address the role of simulation-supported, scenario-based training environments in supporting the development of NDM abilities for integrated business processes.

Naturalistic decision-making is well known from Dr. Gary Klein's study of firefighter chiefs and his development of the recognition-primed decision-making (RPD) model (Klein, 1993). Dr. Klein found that when firefighter chiefs arrive at the scene of a fire, they do not employ a checklist or rationalistic decision-making technique. Instead, they survey the scene, develop a course of action, and execute it. They do this by leveraging their intuition. NDM experts like firefighter chiefs are often not aware of the cues they are considering, how they are weighing them, or the structure of their own mental models which inform their decision-making. Teasing out knowledge about how such experts make decisions and what cues are important for informing their decision-making requires cognitive task analysis techniques which have been developed over the last several decades by cognitive scientists and researchers in such fields as NDM.

The heuristics and biases community provides another important perspective regarding intuition, the skeptical perspective. This community is well known for the work of Dr. Daniel Kahneman and Dr. Amos Tversky, and Kahneman's book *Thinking Fast and Slow*, where they have identified the many limitations of supposed experts in fields such as stock market investing (Kahneman, 2013). Kahneman and Tversky have illustrated how simple statistical techniques can often outperform expert intuition in various domains. In 2009, Kahneman and Klein came together to see whether they could find common ground between their two communities (Kahneman & Klein, 2009). They found that they agreed that reliable intuition could be developed where two factors exist. First, the decision-making environment must have an adequately consistent underlying structure with cues available to inform the decision-maker about the nature of that underlying structure and dynamics impacting their decision-making. Second, the decision-maker must have adequate exposure to the environment, and feedback from it, to inform the development of their mental model relative to the relevant environmental cues.

It is well accepted among NDM and training researchers that scenario-based training can support the development of intuition for NDM environments, facilitating exposure to the environments and feedback for trainee interactions with those environments (Cannon-Bowers et al., 1992; Salas et al., 2006). The development of IS user competencies requires scenario-based training environments which simulate the capabilities and limitations of ISs in the context of the operating environment and organizational processes in which they will be employed. Unfortunately, IS training often consists of either "buttonology" training (context-independent training on how to navigate system interfaces) or seminar programs teaching six sigma or lean management practices.

An example of a tool often found in business management seminars is the Beer Game (see Figure 1), an analog simulation developed by the Massachusetts Institute of Technology (MIT) for teaching participants about the bull whip effect. The bull whip effect, seen in Figure 2, describes the violent oscillation in inventory levels that is seen across a supply chain resulting from variability in demand. Unfortunately, neither buttonology nor seminar-style approaches provide context-specific training necessary for the development of intuition in the employment of complex business management ISs. The Beer Game can support teaching participants about the bull whip effect, and even about techniques for mitigating its effects, but it does not facilitate training participants to recognize the bull whip effect in the context of their organization's specific processes or business management ISs.

Figure 1. College Students Participate in an Analog Beer Game Simulation.
Source: Murray and Curran (2021).

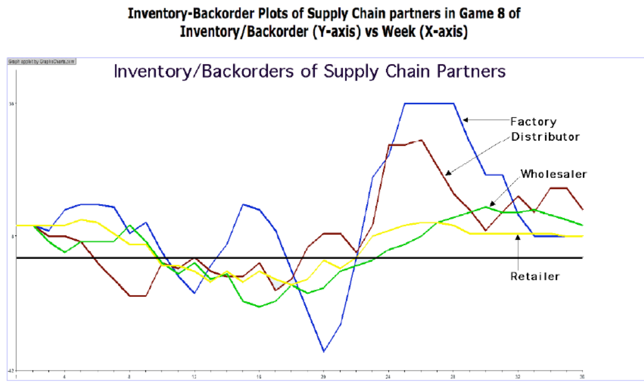


Figure 2. Bull Whip Effect on Inventory/Backorder Levels Across a Supply Chain. Source: Herzog and Katzlinger-Felhofer (2011).

In recent years, commercial interests have registered a demand signal to business schools for graduates who have more than a theoretical understanding of how to leverage complex business management ISs (e.g., enterprise resource planning [ERP] tools) in context. This has resulted in the development of some simulation-supported education tools which develop participants' understanding of how to interpret and respond to business management phenomena through the lens of complex business management ISs. While some of these tools use real world business management ISs (ERPSim uses SAP [Systems, Applications, Products in Data Processing] software as the ERP interface), they are still not adequate for developing intuition, as they are not tailored to organization-specific processes and IS interfaces. While the Department of Defense (DOD) simulations community has succeeded in solving this problem of representing ISs in context for some command and control (C2) systems, many logistics and manpower management systems remain underrepresented in DOD staff training environments.

Despite the DOD's long history of using simulation-supported, scenario-based training environments to support the development of new technology and processes and training for unit commanders and staffs, logistics management ISs are rarely supported. While C2 systems (e.g., blue force tracker [BFT], C2 personal computer [C2PC]) are often included in staff training environments, existing simulation-C2 (sim-C2) interoperability approaches are inadequate for supporting HitL ISs (e.g., logistics information systems, manpower management systems) and representing important dynamics of sociotechnical systems (STS) (e.g., information latency, human error). Including such HitL ISs in staff training environments is essential for developing an appropriate understanding of how to leverage them in context. The representation of STS dynamics is necessary for the development of user understanding of IS limitations and the holistic evaluation of joint cognitive systems (JCSs). This research seeks to overcome the limitations in sim-IS interoperability by developing and evaluating a new approach for sim-IS information exchange through RPA technology.

Addressing the simulation of integrated business processes in context requires more than just a new type of middleware for addressing the lower levels of simulation interoperability. This research approaches the design and development of sim-IS environments as holistic simulations of STSs. Many combat models explicitly model the ability of units or entities to perceive the location and/or status of friendly and enemy units (Tolk, 2012). This modeling of sensing differentiates between the simulation's ground truth and the simulated entities' perceived truth based on the capabilities and limitations of the sensors employed. The simulation of sensor capabilities and limitations, however, has been largely limited to consideration of capabilities and limitations of physical sensors (e.g., thermal sensors, optical sensors), be they digital devices or the human eye. Communications effects servers, which can be used to degrade information exchanged between simulations and C2 systems, simulate network dynamics for communication systems (Bailey et al., 2004). They do not simulate IS dynamics including the nature of ISs in integrated business processes. Sociotechnical systems, those systems consisting of information systems and human operators and staffs, are also sensors which provide information on the statuses of friendly forces (e.g., logistics information, manpower statuses). The simulation of integrated business processes requires treatment of sim-IS information exchange mechanisms themselves as simulations for components of the target STSs being simulated.

A. PROBLEM SPACE

In *The Art of War*, Chinese strategist Sun Tzu asserted that knowing the status of both the enemy and one's own forces is critical for success in battle. Sim-C2 environments which filter ground truth to simulate the range and sensitivity of sensors prepare decision-makers to operate with limited knowledge of their enemy. Sim-IS training environments which do not filter simulation ground truth for friendly forces information misrepresent the information decision-makers can expect to have available regarding the status of their own forces. An example of this is seen with the Joint Deployment Logistics Model (JDLM), which communicated the simulation's ground truth information to the United States (U.S.) Army's Battle Command Sustainment Support System (BCS3) during exercises (B. Nase, U. Larry, & R. Bauer, personal communication, August 2, 2019).

When JDLM provided simulated units' logistics status updates to BCS3, the reports were always current and accurately reflected the ground truth JDLM information (E. Stolle, personal communication, August 19, 2019). The nature of JDLM-BCS3 integration meant that Army logistics units conducted staff training with perfect knowledge of logistics statuses for adjacent and supported units. When they went to the field, however, these units were faced with the challenges of erroneous and frequently delayed logistics status reports (E. Stolle, personal communication, August 19, 2019), a much different (naturalistic) decision-making environment than what they encountered during JDLM-supported training. Information latency and degradation are important factors in many supply chains and are important considerations in leveraging ISs in support of decision-making (Cundius & Alt, 2013; Shattuck & Miller, 2006). Information latency in the propagation of information across integrated business processes can result from the business process, IS, or both. Information degradation can result from human error at any stage in a business process (e.g., entering information into an IS graphical user interface [GUI]) or from limitations in other sensors or the IS itself (Shattuck & Miller, 2006). Neither of these factors is simulated by current sim-IS information exchange approaches.

A related challenge in sim-IS interoperability is a lack of standardized protocols and middleware for automation of sim-IS information exchange (e.g., the Marine Air Ground Task Force [MAGTF] Tactical Warfare Simulation [MTWS] does not support automated information exchange with any logistics information systems). Unlike many C2 systems, which can interoperate through standardized information exchange protocols, information exchange with HitL ISs is often only achievable through manual information exchange (i.e., human simulation operators known as “pucksters”) or through point-to-point engineering (e.g., JDLM-BCS3 interoperability). In addition to being inefficient means of integration, both of these approaches are ill suited for simulating the dynamics of BP-IS integration. A sim-IS interoperability approach is needed which supports automated sim-IS information exchange and the simulation of relevant sociotechnical dynamics to develop trainees’ understanding of IS capabilities and limitations in context.

RPA technology presents an opportunity for addressing these limitations in sim-IS interoperability. Commonly found in commercial off the shelf software from companies such as UiPath, Automation Anywhere, and Blue Prism, RPA technology supports the automation of integrated business processes by replicating human operator interactions with information systems through graphical user interfaces (Miers et al., 2019). This “outside-in” approach to information system integration mitigates the need for engineering changes in operational information systems (or simulations) to support integration of information systems. Instead, existing HitL processes are modeled and automated through a combination of tools such as screen scraping and virtual keyboard/mouse actions.

This research explores the potential for an RPA-based approach to sim-IS information exchange and presents recommendations for a sim-IS Distributed Simulation Engineering and Execution Process (DSEEP) overlay to guide the design and development of sim-IS environments. An RPA-based sim-IS information exchange prototype was developed and evaluated in the context of two simulated sim-IS environments. It was tested for its capacity to support automated sim-IS information exchange and the degree to which such an outside-in approach can simulate the way information delay and degradation manifests in sociotechnical systems. Verification and validation of the RPA-based sim-IS information exchange prototype were conducted to determine the potential for an RPA-based approach to support automated sim-IS information exchange and the degree to which such an outside-in approach can simulate the way information delay and degradation manifest in sociotechnical systems. Recommendations for a sim-IS DSEEP overlay recommendations were compiled concurrently with the design, development, verification, and validation of the prototype architecture to provide guidance for the design and development of sim-IS environments. The recommendations emphasize the role of sim-IS environment conceptual models in the design and development of sim-Is environments and in the design and development of RPA workflows for RPA-based sim-IS information exchange.

1. Information Exchange in Sim-IS Environments

Existing sim-IS information exchange approaches are inefficient and ineffective for supporting the sim-IS environments required for modern staff training in sociotechnical environments. They are cost-prohibitive for including multiple Human-in-the-Loop (HitL) ISs and adapting to the modernization of such systems over time. With their focus on supporting technical and semantic interoperability, they are ineffective for representing the delay and degradation of information propagating through sociotechnical systems. A new means of sim-IS information exchange is needed which supports both the inclusion of more HitL ISs and greater fidelity in sim-IS environment representation of sociotechnical system dynamics.

The exchange of information between simulations and information systems is accomplished through three primary means: “pucksters,” “ad hoc, point-to-point” interoperability (Pullen & Mevassvik, 2016, p. 2), and standards-based interoperability (e.g., Battle Management Language [BML], C2 to Simulation Interoperability [C2SIM]). The focus of sim-IS interoperability research has been on supporting the development of simulation interoperability standards with C2 systems (sim-C2), due to manpower and cost limitations of the first two approaches. C2 systems generally include map-based information systems which support the communication of unit/entity locations and free-text reports for communicating unit statuses and issuing orders through standard protocols. C2 systems are an important subset of the information systems leveraged in modern military staffs, providing commanders and staffs information on the positions and statuses of their units while facilitating some communication between units; however, many more information systems (e.g., logistics and manpower ISs) are relied upon by commanders and staffs.

While the standards-based sim-C2 interoperability approach has been beneficial for integrating simulations with many C2 systems, it also leaves many other information systems unsupported. Sim-C2 interoperability alone is insufficient for supporting training for many staff processes. This research focuses on logistics staff processes (e.g., equipment and supply reporting, maintenance management, personnel casualty treatment and evacuation), though administrative functions are also under-represented in staff training environments. These limitations are evidenced by the many simulation-based training environments which still require a large cadre of pucksters to manually exchange information between constructive simulations and operational information systems unsupported by sim-C2 standards for automated information exchange. Many of these unsupported information systems are HitL ISs, systems that receive their information from human operators manually transferring information in an operational environment vice digital sensors. Such HitL ISs are often also web-based systems lacking the sort of standardized information exchange protocols found with C2 systems. Examples include logistics management systems (e.g., common logistics command and control system [CLC2S], transportation capacity planning tool [TCPT]) and personnel casualty reporting systems (Defense Casualty Information Processing System [DCIPS)]). The high manpower costs and complexity associated with populating these HitL ISs in staff training exercises and wargames often result in only a small subset of relevant systems being represented in the training environment. Conventional sim-IS information exchange approaches also focus on technical and semantic interoperability, with insufficient effort applied to ensuring information exchanged adequately simulates the dynamics of information exchange in sociotechnical systems (e.g., information latency and degradation resulting from human and technological limitations).

2. Sim-C2 Interoperability Approaches

The sociotechnical processes which populate HitL ISs are often simulated by pucksters who manually transfer "ground truth" simulation data into live client terminals for the HitL ISs. Puckster information exchange procedures are rarely, if ever, designed to reflect the real-world state of business process-IS integration, including information latency and degradation. This approach is manpower intensive, limiting the number of ISs which are represented in training environments, and often yields an unrealistic representation of HitL IS capabilities and limitations in context.

Engineering ad hoc, point-to-point information exchange between a simulation and IS decreases the manpower requirements for exercise execution but presents unique challenges for the simulation of integrated business processes. The immediate exchange of information between simulations and operational information systems, as seen with JDLM-BCS3 interoperability, presents an equally unrealistic representation of HitL IS capabilities in the context of the associated business processes. The point-to-point information exchange approach also requires engineering changes to at least one of the systems. This is expensive, in time and money, and is generally inflexible for responding to changes in information systems, processes, and simulations over time.

Many C2 systems are populated using standard data exchange protocols. This approach works for C2 systems which are populated directly by other information systems (e.g., BFT, C2PC) with information like unit locations. Several standards have been developed to support this approach to sim-C2 interoperability, including Coalition-Battle Management Language (C-ML), Military Scenario Definition Language (MSDL), and the C2SIM standard. While this approach is effective for representing the exchange of data between C2 systems, it is poorly suited for supporting the sociotechnical processes by which information is entered into HitL ISs in context.

The technical limitations of current sim-C2 approaches limit the number of HitL ISs which are represented in staff training exercises. Current sim-C2 information exchange approaches also limit the ability of sim-IS environments to support trainees' development of expertise in the employment of ISs in context and development of appropriate levels of trust in their systems.

3. Defining Layers of Sim-IS Information Exchange

In the design of sim-C2 environments, the information exchanged from a simulation to a live C2 system can be conceptualized as an exchange of information from a simulated C2 system within the simulated environment to a live C2 system operated by the training audience (Hamill et al., 2001; Ressler et al., 1999). This exchange of information between the simulated environment and the live C2 systems (those used by the training audience) has three layers:

1. a simulated sensor (digital or human) acquires information about the simulated environment (the simulation's "ground truth") based on the sensor's capabilities
2. the simulated sensor transfers this information to the simulated C2 system
3. the simulated C2 system sends a message to the live C2 system

This is generally implicit in the design of sim-C2 information exchange, and the first two layers (a sensor acquiring information about the simulated environment and sending that information to a C2 system) are underrepresented in the design of sim-C2 information exchanges. C2-Sim interoperability efforts have thus far focused on achieving technical and semantic interoperability between simulations and C2 systems, leaving the simulated C2 systems and simulated sensors to the simulations. Where C2 system stimulation research has addressed the first two layers, the focus has remained on ensuring the technical capabilities and limitations of the C2 systems are represented, without consideration of the human/organizational processes which populate HitL ISs (Hamill et al., 2001). Similarly, while some simulations represent the capabilities and limitations of sensors (i.e., range of a radar system), most simulations do not represent the effects of sociotechnical processes associated with the manual entry of information into HitL ISs.

The default approach for sim-C2 information exchange is a transfer of simulation "ground truth" information (e.g., unit location) from the simulation to the C2 system without delay or degradation, see Figure 3. Global Positioning System-supported location tracking in C2 systems provides an example of how this misrepresents the propagation of information in live systems. Global Positioning System (GPS) sensors are known to have variability of a couple meters, with a tendency to "jump around" (NATO Science and Technology Organization, 2015a, p. A-58). This is the sort of information degradation addressed by the Dynamic Model of Situated Cognition (DMSC), discussed in Chapter II. When a simulation populates a live C2 system with simulated unit tracks, however, simulated GPS sensor reports for unit locations are exact and consistent. No effort is made to simulate the GPS signal variability. For C2 systems, this is generally a reasonable approach, as unit location disparities of a couple meters are rarely relevant to a commander's decision-making. For some C2 processes, however, **this approach to sim-C2 integration can present an important misrepresentation of C2 capabilities in context**.

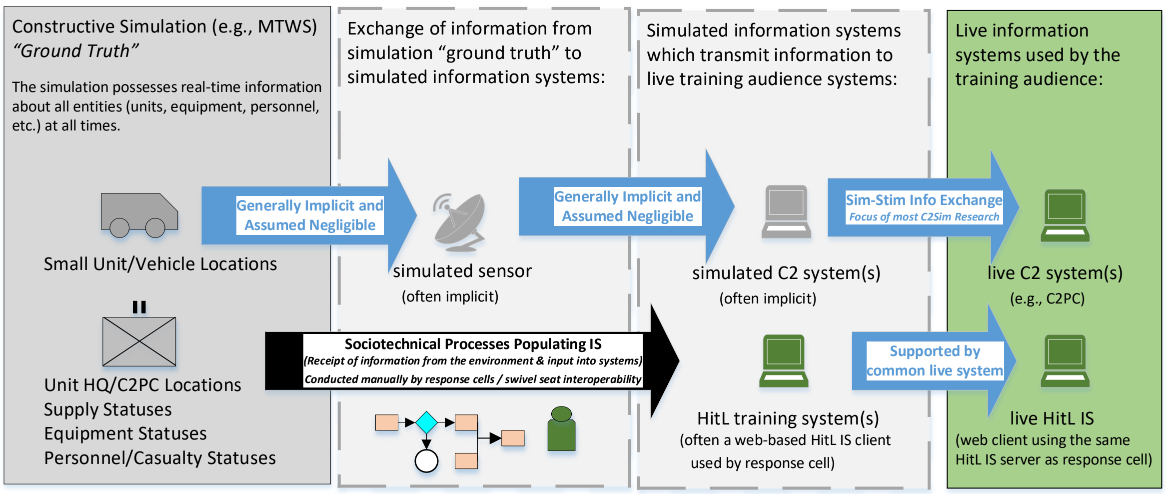


Figure 3. The Layers of Sim-IS Interoperability.

Unit locations and statuses are sometimes communicated between units via radio, with a recipient manually updating the digital C2 system to reflect the updated information for the subordinate unit or convoy. Such a process would result in unit locations being updated infrequently in C2 systems with the accuracy of a reported location depending on the sender's map-reading abilities and the time it took for the recipient to receive the information and manually populate the digital C2 system. In these instances, ***a failure to represent the delay and degradation of information populating C2 systems can yield negative training, with trainees developing too much trust in the accuracy and timeliness of the information presented in their information systems.*** For logistics processes, the focus of this research, these effects can be even greater.

4. Limitations of Current Interoperability Approaches for Sim-IS Environments

For HitL ISs, the first two layers of sim-IS information exchange can generally be expected to impose a much greater impact on the accuracy and timeliness of the information exchanged with the live system. While C2 system information is often generated by digital sensors with known ranges and sensitivity, real-world HitL ISs are populated by more fluid, imprecise sociotechnical processes. These processes are often rife with variable time delays of information, human error, and biased perspectives based on how the sending and receiving organizations have integrated the information systems in their processes.

Whether sim-IS information exchange results from pucksters or automated point-to-point system integration, a failure to represent the processes which populate the ISs results in a misrepresentation of information systems capabilities for the training audience. It is often unrealistic to expect a simulation (or sim-IS information exchange mechanism) to simulate such sociotechnical processes themselves. Organizational processes and integrated information systems constantly coevolve, and simulating them would be quite difficult and resource intensive. Some of the effects of such sociotechnical processes, such as time delay and variability of information propagating through the STS can, however, be measured and represented in sim-IS information exchanges.

Unit fuel statuses reporting processes provide an example of how the role of ISs in sociotechnical processes can impact the information represented in HitL ISs. Constructive simulations often track and report unit fuel levels as aggregate quantities for a unit. If a puckster were to retrieve such a fuel report from the MAGTF Tactical Warfare Simulation (MTWS), for example, and enter said information into the appropriate HitL IS client, the training audience would receive a much more accurate representation of unit fuel levels than they could realistically expect in the real-world. This is a current problem being faced within the Marine Corps staff training community by organizations such as the MAGTF Staff Training Program (MSTP) and Marine Corps Logistics Operations Group (MCLOG). The current puckster-based sim-IS information exchange approaches for converting MTWS fuel reports to bulk fuel estimates are very manpower intensive and do not represent the variability in report accuracy that occur in actual operations. This issue is illustrated in the following example.

In real-world operations, logisticians are only able to report the fuel available within bulk fuel storage containers. The fuel statuses of individual trucks and generators are unknown, as it is not feasible to check all individual fuel tank levels for each logistics status report. As illustrated in Figure 4, at the time fuel levels are reported, the fleet of vehicles may have full gas tanks (i.e., 4,060 gallons) or they may be returning from a long convoy with less-than-full gas tanks (i.e., 3,280 gallons). Commanders and staffs rarely receive fully accurate and timely information regarding unit statuses, particularly for those pieces of information that propagate through lengthy sociotechnical processes where the effects of information latency and degradation accrue. It is important for the decision-makers and staffs who leverage ISs to understand the capabilities and limitations of those systems and the processes which populate them. Opportunities to learn these lessons, and exercise procedures to deal with the attendant issues, are not available in conventional sim-IS training environments.

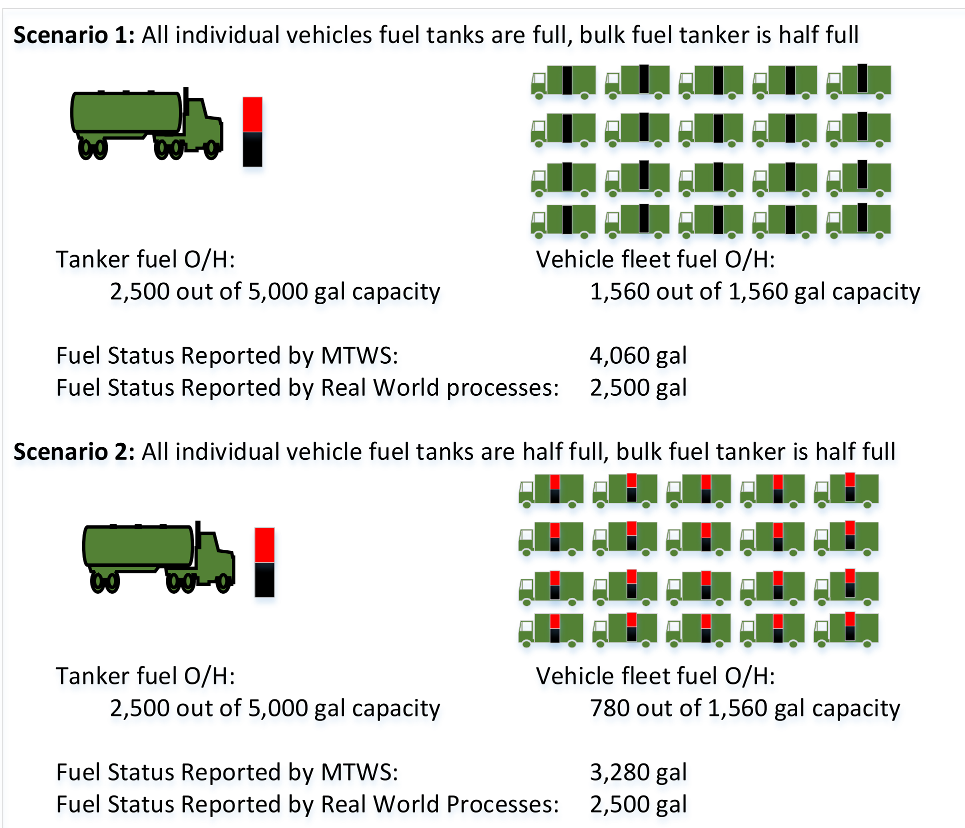


Figure 4. The MTWS Fuel Reporting Challenge.

The tight coupling of JDLM and BCS3 illustrates the limitations of ad hoc, point-to-point sim-IS interoperability. JDLM (a constructive logistics training simulation) and BCS3 (the Army's former logistics information system) were developed to be tightly coupled in support of scenario-based staff training. This point-to-point interoperability between JDLM and BCS3 ensured that BCS3 accurately represented the consumption, transportation, and distribution of various supplies and equipment simulated within the JDLM simulation throughout the staff training exercises. While this facilitated staff training with regard to BCS3 functionality and internal procedures, it also resulted in a high level of trust in the timeliness and accuracy of the system. When units conducted subsequent field training in which personnel, not a simulation, populated BCS3, they realized that they had developed too high a level of trust in the timeliness and accuracy of the system and the processes through which it is populated in real-world operations (E. Stolle, personal communication, August 19, 2019). In 2016, the U.S. Army Combined Arms Support Command acknowledged these training limitations in their "Sustainment Constructive Training White Paper":

> The principle of unit training that is currently most misaligned for sustainment units in Live, Virtual, Constructive and Gaming (LVC-G) exercises today is the "Train as you will fight" principle. This principle means that units, Soldiers, and leaders train under an expected operational environment for their assigned mission. The relevant information required to train units on key material and distribution management is there, but doesn't properly replicate how the information is provided by the information systems used in operations. (United States Army Combined Arms Support Command, 2016a, p. 5)

It is unrealistic to expect constructive simulations to reflect the dynamics of information propagation through sociotechnical systems, considering the multiple information systems with which they may be coupled. Those information exchange dynamics depend on the information system being populated, and the processes will evolve over time, possibly also varying between units. The burden for effectively simulating the exchange of information between the simulation's "ground truth" and the trainee's IS falls upon the sim-IS environment designers and developers. Designers and developers require new tools to design and build sim-IS environments.

B. SIM-IS ENVIRONMENTS FOR JOINT COGNITIVE SYSTEM COEVOLUTION

Sim-IS environments are also important for the development of information systems and associated processes as joint cognitive systems. The coevolution of business processes, information systems, and human competencies requires the feedback that comes from operating in context. It also requires representation of the dynamics of socio-technical systems which make decision-making and collaborative work challenging (e.g., information latency, data errors). In the mid-20th Century, the German military conducted "radio exercises" to better understand how a new technology, radios, could support blitzkrieg (Citino, 1999). Today, simulation-supported scenario-based environments are necessary for supporting not only training but also the holistic development of information technology and the processes through which it is employed. While this research is focused on the potential for RPA-based sim-IS environments to support training, it also presents environments to support wargaming and integrated business process coevolution.

Exploiting Kasparov's Law and enhancing the effectiveness and efficiency of integrated business processes requires the continuous development and coevolution of the operating environment (e.g., integrated business processes), ISs, and user competencies. These three factors are referred to as the elements of the cognitive triad of joint cognitive systems in the field of cognitive systems engineering (CSE) (Eggleston, 2002; D. D. Woods & Roth, 1988). CSE researchers emphasize the importance of taking a holistic approach to JCS design and continuous improvement, as "inattention to any element can produce a human-computer system that fails" (Roth et al., 2000, p. 7). Scenario-based training and evaluation environments are particularly important for supporting the holistic development of JCSs, including the development of user competencies.

The continuous coevolution of joint cognitive systems is pursued in the CSE community through simulation environments such as synthetic task environments, scaled-worlds, micro-worlds, and spartan labs (M. J. Miller & Feigh, 2019). CSE approaches for evaluating JCS's often require the use of these environments, though the environments built in support of CSE studies are usually not persistent environments intended to support continuous, cyclical development across all elements of the JCS. Such persistent environments would need to support personnel and staff training, evaluation and reengineering of organizational processes, and analysis of ISs for coevolution with organizational processes and user competencies. Support for these holistic JCS-oriented applications is a potential application for the sim-IS environment architecture presented in this work, worthy of future research.

C. RESEARCH QUESTIONS AND METHODOLOGY

This research provides insight into the potential for RPA to facilitate an "outside-in" approach to sim-IS interoperability. This includes exploration of the following research questions:

1. **Primary:** How can an RPA-based "outside-in" approach to sim-IS information exchange simulate temporal and content quality degradation of information across sociotechnical systems?
2. Subsidiary: How can an RPA-based "outside-in" approach to sim-IS information exchange facilitate automated exchange of information between simulations and HitL ISs in support of training exercises?
3. Subsidiary: How can RPA-based middleware and data exchange models for sim-IS information exchange be designed to facilitate modularity and reuse across different sim-IS environments?
4. Subsidiary: How can sim-IS environments be designed to address the dynamics of BP-IS integration in sociotechnical systems?

Research questions 1 and 2 are addressed through the design, development, verification, and validation of a prototype RPA architecture for sim-IS information exchange and simulation of STS information exchange dynamics. Verification of the RPA architecture is supported by simulation of the prototype architecture across four experiments, with quantitative analysis of the sim-IS information exchange and the simulation of specified STS information exchange dynamics. This quantitative analysis of sim-IS information exchange and simulation of STS dynamics is necessary but not sufficient for the validation of the RPA architecture. Validation of the RPA-based sim-IS information exchange architecture is achieved by supplementing the quantitative analysis with qualitative analysis in the form of subject matter expert interviews following a field demonstration of the RPA prototype conducted for the research sponsor aboard the Marine Corps Air Ground Combat Center. Research questions 3 and 4 are addressed through informed argument. Insights gained in the design, development, verification, and validation of the RPA-based sim-IS architecture inform recommendations for an envisioned DSEEP overlay for RPA-based sim-IS environments.

D. DISSERTATION ORGANIZATION

This research employs a design science research (DSR) approach to explore the potential for RPA technology to support sim-IS environments with the simulation of integrated business processes and associated STS dynamics. Chapter II presents a review of literature relevant to this research. The research methodology is presented in Chapter III along with the research hypotheses for each of the research questions. This includes an introduction to the DSR approach and discussion of how the development and evaluation of DSR artifacts (instantiation, method, model, or construct artifacts) supports the extension of knowledge within the DSR paradigm. For this research two DSR artifacts are presented.

> Design science…creates and evaluates IT artifacts intended to solve identified organizational problems. Such artifacts are represented in a structured form that may vary from software, formal logic, and rigorous mathematics to informal natural language descriptions…As field studies enable behavioral-science researchers to understand organizational phenomena in context, the process of constructing and exercising innovative IT artifacts enables [sic] design-science researchers to understand the problem addressed by the artifact and the feasibility of their approach to its solution. (Hevner et al., 2004, p. 77)

The first DSR artifact developed in support of this research, an instantiation artifact, is the prototype RPA-based sim-IS information exchange architecture and RPA modules for two RPA-based sim-IS environments. A description of the RPA-based sim-IS information exchange architecture developed in support of this research, and the prototype modules for the two sim-IS environments, is presented in Chapter IV. Chapter V describes how verification of the RPA-based sim-IS information exchange architecture was conducted through simulation of STS dynamics across two sim-IS environments using different constructive simulations integrated with the CLC2S logistics IS using the prototype RPA-based sim-IS information exchange architecture. The verification of the RPA-based sim-IS information exchange architecture is necessary, but insufficient for validation of the architecture. Chapter VI presents the results of the demonstration of the architecture the research sponsors aboard Marine Corps Air Ground Combat Center, in support of artifact's validation.

The second DSR artifact developed in support of this research, a method artifact, consists of recommendations for an overlay to the DSEEP for RPA-based sim-IS environments. Throughout this research, numerous lessons were learned regarding considerations that must be taken at various stages in the design and development of RPA-based sim-IS environments. Chapter VII presents recommendations for how these issues may be addressed in a DSEEP overlay for RPA-based sim-IS environments. This dissertation constitutes the first steps in exploring and designing an RPA-based approach to sim-IS information exchange with the simulation of integrated business processes and associated STS dynamics. There are many considerations deserving of future work. Chapter VIII presents a summary of the research contributions and identifies several issues to be addressed in future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

Military services rely on live-virtual-constructive environments to support training in tasks and situations which are infrequent and too expensive or dangerous to replicate in a fully live environment. Modern operations centers are critical for the effective management of military units in the field and emergency operations management for natural disasters and other humanitarian crises. In the Marine Corps and Army these are commonly found in the form of combat operations centers (COC). These operations centers often include cross-functional teams leveraging complex information systems to support decision making in time-constrained, high risk decision-making environments. Military operations occur infrequently, and COCs are often manned by personnel whose only COC experience comes from training exercises. In the DOD, these exercises can range from live field exercises to constructive simulation-supported training environments.

Modern operations centers require personnel who effectively leverage information systems to support decision making and coordination. Staff training exercises provide opportunities for units to exercise and evaluate their staff processes, including an evaluation of how effectively the processes leverage information systems to support collective work and decision making. These exercises also present decision makers opportunities to exercise their decision-making skills, including their ability to leverage information systems to inform their decision making. Unfortunately, existing sim-IS environments are insufficient in their representation of all relevant information systems (e.g., logistics and administrative systems).

Extensive research in the fields of training and naturalistic decision-making has demonstrated the importance of scenario-based training environments for developing staff members' mental models regarding the sociotechnical systems where they operate. Information system user training which fails to address the context of IS implementation is insufficient. While decision-making theory and training principles exist which can guide the design of scenario-based training (SBT) environments, obstacles remain which preclude the development of simulation supported SBT environments for HitL information systems and their associated business processes.

Simulation-supported SBT within the armed forces is often conducted through the use of distributed simulation environments. The integration of different simulations (e.g., flight simulators, constructive battlefield simulations, virtual convoy simulators) facilitates the design and development of training environments. The design and development of these distributed simulations is accomplished through the use numerous standards. The DSEEP standard (Institute of Electrical and Electronics Engineers (IEEE) Standard 1730–2010) guides the design and development of distributed simulation environments (IEEE Std 1730, 2011). Depending on the training requirements and available simulations, standards also exist to guide the implementation of different protocols (e.g., High Level Architecture, Distributed Interactive Simulation) and the documentation rules established for the distributed simulation environment (e.g., Federation Engineering Agreements Template). While DSEEP overlays exist to address the unique considerations for the use of different distributed simulation architectures (or multiple architectures) insufficient guidance exists for the development of sim-IS environments. DSEEP does not provide guidance regarding the design and development of sim-IS environments which simulate the dynamics of BP-IS integration in sociotechnical systems. This is partly due to the absence of sim-IS information exchange mechanisms and standards to address these challenges.

The dynamics of sociotechnical systems, such as information latency and degradation, are not represented in sim-IS training environments and are not supported by existing simulation interoperability standards. While such dynamics could potentially be simulated through pucksters and point-to-point interoperability approaches, these methods are too expensive in time, money, and manpower to support the integration of simulations and the various HitL information systems leveraged by modern staffs. Robotic process automation technology has potential for supporting the integration of simulations and HitL ISs in a way which represents BP-IS dynamics. RPA is a type of software which has been used to automate BP-IS integration and support information exchange between different operational information systems. While it has been primarily employed in support of back office processes (e.g., financial audits and report generation), the present research shows that this low-cost, outside-in approach to process automation and IS integration has the potential to support a new means of automated sim-IS information exchange.

A. SIM-IS TRAINING REQUIREMENTS FOR INTEGRATED BUSINESS PROCESSES

In modern work environments, remaining competitive often depends on an organization's effective use of complex information systems (e.g., cross functional enterprise resource planning (ERP) systems) which support collaboration across multidisciplinary staffs. Effective use of such information systems requires the alignment of business processes and information technology and a work force which understands how to leverage information systems in the context of the business processes. The former is accomplished through iterative business process redesign and information system changes as the processes and information systems coevolve. The latter requires a continuous development and refinement of worker competencies for the ever-changing integrated business processes. Both can benefit from simulation-supported environments.

Achieving both business process-information technology alignment (BITA) and a well-trained workforce has proven challenging for most organizations and both have been cited as leading causes of failure in the fielding of enterprise information systems (Alcivar & Abad, 2016; Caya et al., 2014). These challenges are also closely related, as the coevolving business processes and information systems require training environments which evolve with them (Al-Mashari et al., 2003; Hoffman & Fiore, 2012). Increasing demand for business managers with more comprehensive ERP competencies has resulted in the development of new training and education techniques and technology for developing integrated business system conceptual knowledge and transactional ERP system skills (Charland et al., 2015; Nisula & Pekkola, 2017).

These experiential learning environments are tools to support education. They have proven effective for the development of general ERP competencies and knowledge regarding the management of integrated business processes. However, they are not training tools for the development of naturalistic decision-making abilities in context. Existing ERP-centric experiential learning simulations, like ERPSim, are not tailored for specific organizations and are not intended to support training for personnel on organization-specific information systems and processes. The primary method for training personnel to leverage information systems in a business context remains on-the-job training. Although this may be acceptable for organizations which exercise their integrated business processes daily and can afford to train personnel through on-the-job training, it is insufficient for those organizations whose operations are infrequent and high risk.

Military and emergency management organizations cannot afford to rely on on-the-job training to develop individuals' competencies and intuition regarding the organizations' integrated business processes (Hannay & Kikke, 2019). In these organizations, scenario-based training is relied upon to develop individual and team competencies for sociotechnical processes that are only employed in times of war or humanitarian disasters. For staffs, such training is conducted in the form of field exercises and simulation-supported staff training exercises. Military staff training exercises include command post exercises (CPXs) addressing C2 procedures and battle staff training exercises (BSTXs), which exercise broader processes, planning, decision-making, and staff coordination. These exercises facilitate both staff training and the evaluation of integrated business processes themselves. As information systems have proliferated within military COCs, significant effort has been exerted to include operational information systems in staff training environments.

While DOD organizations possess extensive simulation capabilities and a wide variety of information systems, no research addresses the integration of HitL ERP systems in such training environments. The integration of military simulations and information systems for staff training exercises has instead centered around C2 systems. These systems have relatively clear roles and methods of employment in support of staff processes, unlike the more complex business process-IS integration challenges found with other information systems.

1. Defining Business Context Knowledge for Integrated Business Processes

Research in information system end-user training provides a broad understanding of the knowledge and skills necessary for the effective use of complex information systems in integrated business processes. Although most end-user training research addresses the development of transactional information system skills, researchers have increasingly identified the importance of cognitive and affective dimensions of performance (Gupta et al., 2010). This research has yielded multiple knowledge frameworks which address the types of business context knowledge necessary for successful use and management of integrated business processes (Bassellier et al., 2001; Gupta et al., 2010; Kang & Santhanam, 2003; Sein et al., 1999). The Model of Collaborative Application Training, for example, defines seven types of knowledge organized across three levels: application knowledge, business context knowledge, and collaborative task knowledge (Kang & Santhanam, 2003); refer to Figure 5.

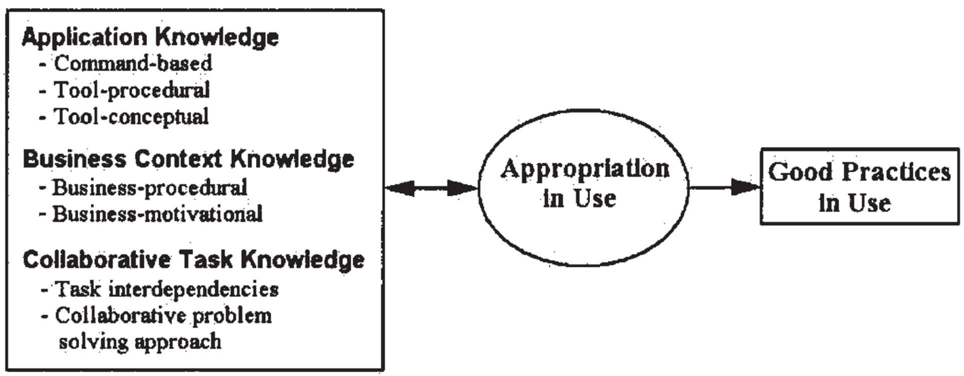


Figure 5. The Model of Collaborative Application Training. Source: Kang and Santhanam (2003).

Business management and information system training have historically been conducted separately, with neither addressing the development of business context knowledge or collaborative task knowledge. Business management analog simulations, such as those found in lean/six sigma courses, provide education regarding general business management concepts. The "beer game," for example, teaches participants about the effect of time delay in the exchange of information across a supply chain, a phenomenon known as the bull whip effect (Tan et al., 2010). The bull whip effect, presented earlier in Figure 2, describes how inventory tends to oscillate violently between high inventory and backlog statuses as different nodes of a supply chain attempt to anticipate future product demand requirements. While the beer game, and similar analog programs, provide valuable education regarding business management processes in generic contexts, it is often left to the students to identify how the concepts apply to their specific organizations' processes and information systems.

At the other end of the education-training spectrum, conventional information system end-user training often supports the development of transactional information system skills without addressing the information system's role in the business process (Gupta et al., 2010). These conventional approaches, skills-based IS end-user training and conceptual business management education, often do not address the integration of information systems in business management.

2. Building ERP Competence through Experiential Learning

As industries demand graduates who are better prepared to manage integrated business process, business management schools have increasingly sought to close the gap between conceptual business management education and transactional end user training (Antonucci et al., 2004; Charland et al., 2015; Leger et al., 2011). This research has yielded experiential learning environments which target the development of the ERP competencies (Charland et al., 2015; Cronan et al., 2012; Nisula, 2019; Nisula & Pekkola, 2017). While these experiential learning environments support the development of some transactional skills and conceptual business management knowledge, they excel at developing students' general understanding of business process-IS integration.

Information technology (IT) competency researchers have described IT competence as consisting of explicit and tacit knowledge regarding the operation of IT and its use in support of business processes. Explicit knowledge consists of "knowledge of technologies, applications, systems development, and management of IT" (Bassellier et al., 2001, p. 159). Bassellier et al. (2001) define tacit IT knowledge as consisting of the individual's experience and their cognition, with cognition consisting of "two mental models: the manager's process view and his or her vision of the role of IT" (Charland et al., 2015, p. 33). While explicit IT knowledge is necessary for the development of tacit IT knowledge, it has been found that explicit knowledge is insufficient for developing ERP competency (Charland et al., 2015, p. 36). One tool for creating experiential learning environments for development of tacit IT knowledge is ERPSim.

ERPSim is a simulation which is tightly coupled with a real-world ERP system, SAP, and challenges students to manage a notional business by interacting directly through the ERP interface. ERPSim simulates generic business processes which are intended to approximate common business processes found in industry. The user interface (UI) is the actual SAP software to support development of students' transactional skills with SAP software commonly found in industry. Other ERP simulations similarly use generic business processes and information systems, with the aim of developing students' understanding of integrated business processes (e.g., SAP ERP software) (Nisula, 2019).

Despite these advances in business management education, a significant gap remains in business management training. ERP simulations and business skills labs are rarely designed to support training for the development of transactional information system skills, business context knowledge, or collaborative task knowledge for a specific organization's integrated business processes. After all, the developers of ERPSim describe it as a "simulation game for teaching ERP concepts" (Leger et al., 2011, p. 41), not as a tool to support training for specific integrated business processes. Even when ERPSim training is presented to experienced workers, vice college students, the training is provided using generic business processes and information system (BP-IS) interfaces for a notional organization (Deranek et al., 2017). While generic business skills labs and ERP simulations support the development of knowledge regarding concepts and integrated business process management approaches, they are not designed to support development of knowledge, or naturalistic decision-making abilities, for specific BP-IS integration within an organization.

The development of transactional IS skills, business context knowledge, and collaborative task knowledge for a particular organization/work environment requires the design of training environments which emulate the organization's unique business processes, information systems, and the integration of those business processes and information systems. ERP simulation and business skills lab research provide valuable insights into the types of knowledge which must be targeted with such training as mental models are developed for specific integrated business processes. However, the development and execution of such organization-specific training environments falls outside the scope of existing information systems research.

3. Simulation-Based Training for Development of Naturalistic Decision-Making

Largely beginning with the Tactical Decision-Making Under Stress (TADMUS) research of the 1990s, research in decision making and training has highlighted the importance of simulation-based training for the development of individual and shared mental models (Cannon-Bowers et al., 1992; Salas et al., 2006, 2009). Educators leverage experiential learning environments to develop declarative knowledge and support a generalized "declarative-to-procedural shift" in the application of such knowledge (Hoffman et al., 2014, p. 23). Training managers use scenario-based training to support this "declarative-to-procedural shift" in knowledge which transfers to a target environment, an automatization of knowledge (where possible), and/or the development of intuition for naturalistic decision-making environments.

> The situation has provided a cue; this cue has given the expert access to information stored in memory, and the information provides the answer. Intuition is nothing more and nothing less than recognition.
>
> – Herbert Simon (Kahneman, 2013, p. 237)

To support the development of intuition, a training environment must present trainees with numerous scenarios which present appropriate cues and feedback for the intended tasks and operating environment. While simulations used for education often leverage generic contexts (such as generic business processes for notional companies), training environments must be tailored to the specific operating environment, systems, and processes where the trainee is expected to operate. The availability and consistency of cues and feedback provided within a target operating environment influence the degree to which training is designed to support proceduralization of knowledge and automaticity (consistent environments/tasks) or the development of intuition (naturalistic environments). In either case, training environments generally designed to prepare trainees for performing within the target operating environment in order to support training transfer. Two particularly relative theories, cognitive flexibility theory and cognitive transformation theory, identify the value of extraneous cognitive load in the design of such training for NDM environments. These theories suggest that artificially decreasing extraneous cognitive load degrades the long term transfer of training, so long as the distractors are a relatively accurate representation of environmental conditions (Hoffman et al., 2014). In the context of integrated business processes, while generic business skills labs may support the development of general knowledge regarding integrated business processes, they do not support proceduralization of application knowledge or the development of intuition regarding the trainee organization's processes. The knowledge gained from business skills labs, or ERPSim, must be followed by on-the-job training in order to develop mental models specific to the integrated business processes of a trainee's organization.

Modern workplaces, which include multi-person staffs with different skill sets and multiple information systems, require the development of both individual and shared mental models. In many cases it would not be feasible for each individual to possess the knowledge required by all of their peers. Instead, training organizations aim to develop "optimally consonant schemas" (Duffy, 1995, p. 349), where team member mental models overlap just enough to ensure effective and efficient team collaboration in support of their assigned mission.

Research in the field of situated cognition has also demonstrated the importance of developing information system-users' understanding of the capabilities and limitations of information systems relative to the sensors and systems that populate said information systems. The Dynamic Model of Situated Cognition (DMSC) illustrates how information is degraded and transformed as it propagates through sensors and information systems to a decision maker (Shattuck & Miller, 2006); see Figure 6. Understanding the limitations of sensors, C2 systems, and the translation of information as it propagates between such systems is important for developing appropriate levels of human-machine trust and interpreting the environment perceived through the technological systems.

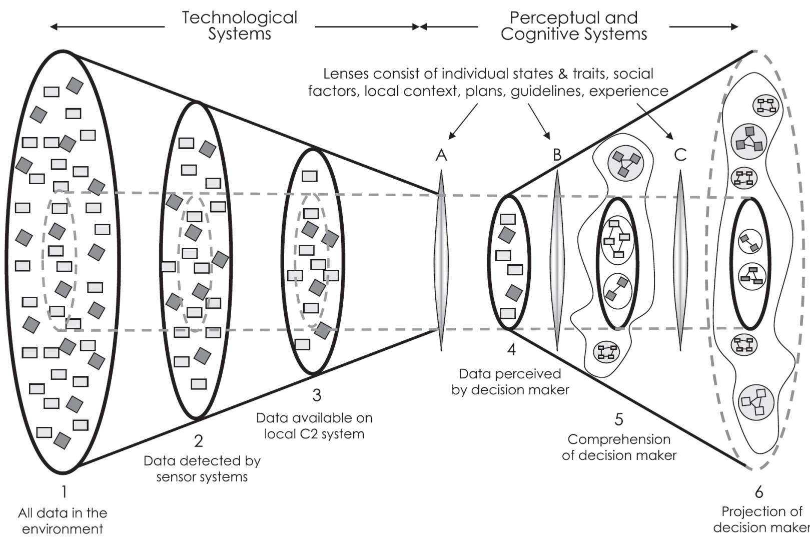


Figure 6. Dynamic Model of Situated Cognition. Source: Shattuck and Miller (2006).

While the DMSC illustrates the degradation of information as it propagates through technological systems, information delay and degradation resulting from sociotechnical systems can be even more significant. In supply chain management, information delays and order variability are important factors which influence manager decision-making (Cundius & Alt, 2013). The bull whip effect, introduced earlier, describes how variability in perceived end-user demand is amplified as it propagates across the supply chain due to information delay and degradation, among other factors. While analog simulations, like the beer game, can educate students about the effects of such factors and resulting phenomenon it is difficult for training organizations to simulate such phenomenon using live information systems. It is notable that the beer game is rarely conducted using real-world information systems and processes, leaving participants to identify how to apply the lessons on their own. Even with tools like ERPSim, students are left to interpret the application of theory to the practical management of their organizations' integrated business processes on their own. Scenario-based training environments are needed which can support the development of business context knowledge for organization-specific integrated business processes. **This requirement is especially pressing for those organizations which infrequently exercise their integrated business processes, such as DOD and emergency management organizations.**

It is important to note that training for integrated business processes is also not only a challenge for new system users or those unfamiliar with an organization's processes. As processes and information systems coevolve, the ERP competencies required of personnel, and the more nuanced mental models about the processes and the nature of BP-IS integration in context, must also change (Hoffman & Fiore, 2012). Development of such mental models and ERP competencies requires training environments which simulate relevant dynamics of integrated business processes.

The value of integrating ERP systems in staff training evolutions is recognized within the DOD. In 2016 the U.S. Army Combined Arms Support Command (CASCOM) highlighted the importance of including ERP systems in unit training events (United States Army Combined Arms Support Command. (2016b). In the same report, however, CASCOM acknowledged that the integration of ERP systems in collective training environments was a significant training capability gap. As of September 2019, the Future Simulations Division Chief for the U.S. Army's leading logistics training design and development organization, Logistics Exercise and Simulation Directorate (LESD), reaffirmed the importance of such training. He asserted that the "ideal solution [for constructive staff training] is to stimulate the 'real-world' systems and the training audience uses these business systems / mission command systems to drive unit information requirements and update estimates" (J. Rauguth, personal communication, September 25, 2019). Furthermore, he confirmed that "most [Army] business systems do not have the ability to be used in a training exercise" (J. Rauguth, personal communication, September 25, 2019).

While individual competencies and the teamwork/task work dynamics of teams have received considerable attention over recent decades, the training community has paid less attention to the holistic development of work systems. New technology is often acquired with an expectation that it will disrupt legacy processes and competency requirements, but without an understanding of how (Klein & Ralls, 1997). This can invert the traditional relationship between job analysis and training, as technology training ends up influencing the way technology is implemented on the job (Klein & Ralls, 1997). In the field of cognitive systems engineering, this dynamic, which challenges system designers to anticipate how systems dynamics and requirements will change following the arrival of new technology, is referred to as the "envisioned world problem" (Woods & Dekker, 2000).

B. JOINT COGNITIVE SYSTEMS

Cognitive systems engineering has been defined as "an approach to the design of technology, training, and processes intended to manage cognitive complexity in sociotechnical systems" (Militello et al., 2009, p. 263). This holistic approach makes CSE methods particularly well suited for the design of training environments for the evaluation of human competencies in the context of ISs and business processes. A primary tenet of CSE is that human-machine systems for cognitive work should be approached holistically as joint cognitive systems, consisting of environmental constraints, the human decision-maker(s), and the technological system(s) (Hollnagel & Woods, 2005; Smith & Hoffman, 2018). These three elements are known as the cognitive systems triad (Woods & Roth, 1988). When modifications are made to any of the three elements of the cognitive systems triad (e.g., developing a new IS or improving an existing one), changes should be expected in the other elements.

Changes in the design of a computerized decision support system (DSS), for example, will likely yield changes to the associated organizational processes and the cognitive competencies and experiences required for users to achieve target performance outcomes. This consideration is particularly relevant for those seeking to increase the role of artificial intelligence in decision support systems. Offloading human tasks to an IS, for example, can result in risks such as user out-of-the-loop-unfamiliarity and inappropriate levels of trust in DSS recommendations, which must then be mitigated through modifications to some or all elements of the JCS. Simulation supported environments facilitate study of the dynamics of current and anticipated work domains and the continuous coevolution of JCSs.

CSE is often mentioned in the development of new technological systems, or modifications of existing systems. CSE methods are equally applicable to the design of training programs or reengineering of organizational processes as they are in supporting the acquisitions process. While this research is focused on supporting the development of sim-IS environments for training purposes, the CSE perspective suggests that training design should be conducted with consideration of the broader JCS. It is therefore reasonable to expect that sim-IS environments for training would also support the holistic evaluation and coevolution of JCSs. This section provides insights into JCS design considerations and the methods for evaluating a JCS in context.

1. Supporting the Human Decision-Making Process

For JCSs supporting a decision-making process, the human's role in assigning value to variables and outcomes cannot be overstated. The human must maintain an understanding of when different decision-making approaches are warranted, an understanding of which variables the computerized IS is, and is not considering, and how the IS is considering the variables. That said, although "the quality of a decision may rest less on the process of choice and more on the process of assigning value," the process of choice is still important, and the relative value between the two processes ultimately depends on the decision-making task at hand and the needs of the decision-maker (Flach & Voorhorst, 2020a, p. 222). The front-end analysis will provide insight into those needs and help prioritize efforts in the design of the decision-making aid. This section addresses some considerations for the design of JCSs to support the human decision-making process and identifies some common challenges.

a. Balancing an Ecological and Analytical Approach

One way to categorize ISs is the degree to which they support recognition of the environment versus support cognitive tasks through semi-independent analysis and decision recommendations. Considering ecological and analytical perspectives can aid in ensuring both perspectives are addressed, but this should not be approached as an either-or situation. Both perspectives are necessary for the design of an intuitive, effective IS.

The ecological approach to cognitively complex domains suggests that rather than designing decision-making aids which augment the decision maker's analysis through normative decision-making algorithms and parallel analysis, decision-making aids should support the decision maker's interpretation of the environment's 'deep structure', to augment the “recognition processes” (Flach & Voorhorst, 2020a, p. 177). This approach seeks to enhance the human decision-maker’s ability to recognize patterns (i.e., recognition primed decision-making), recognizing that the human is best suited to evaluate “the values (i.e., the costs and benefits) associated with various options and consequences” in the decision-making environment (Flach & Voorhorst, 2020a, p. 222).

While the ecological interface design (EID) approach focuses on designing the decision-making aid to improve the decision-maker’s understanding of the environment, the analytical approach focuses on assisting the decision-maker in resolving the cognitively complex tasks identified with cognitive task analysis (CTA) methods. Decision support systems are often designed to support human decision-making in well-defined decision making situations which exceed human analytical capacity. Artificial intelligence-enabled machines can aid human decision-makers by applying robust, normative algorithms to provide insights into situations which do not support intuitive expertise or which exceed the analytical capabilities of the given operator.

Components of a decision-making environment which can be reinforced through analytical decision making aid elements can be identified through an exploration of decision-making tasks and sub-tasks along the lines of the machines-are-good-at versus humans-are-good-at (MAGA-HAGA) list developed by human factors psychologists such as Paul Fitts (Flach & Hoffman, 2012; Hoffman et al., 2012). This approach would identify, for example, that humans are generally better at assigning the values to variables and outcomes in a fluid environment, such as weighting the importance of efficiency or risk-to-force against risk-to-mission. AI-enabled (artificial intelligence) machines, meanwhile, are generally better suited for conducting complex calculations based in a highly structured environment such as anticipating maintenance requirements for routine missions. The decision-making aid can therefore be designed to provide analysis of the well-structured elements of the environment, allowing the decision-maker to focus on analysis evaluating the ill-defined elements better supported by intuitive expertise.

While it may provide some helpful generalizations, the MAGA-HAGA approach has also been criticized as oversimplifying human capabilities and limitations, as well as the dynamics of human-machine integration (Hoffman et al., 2012; Woods, 2002). These

MAGA-HAGA distinctions, as represented in Paul Fitts's widely referenced "Fitts List," are complicated by the variables such as the ever changing dynamics of ill-defined decision-making environments and the varying degrees of human expertise. David Woods, a leader in the field of CSE, provides an alternative approach by recommending the use of an "Un-Fitts List," which "presents a rich view, one that does not focus on human shortcomings" (Hoffman et al., 2012, p. 110). Taking this approach and applying the associated Human-Centered Computing (HCC) principles of system design (e.g., the Aretha Franklin principle) will aid the designer in balancing consideration of human information processing capabilities and environmental constraints (Flach & Dominguez, 1995).

> The Aretha Franklin Principle of HCC: Do not devalue the human in order to justify the machine. Do not criticize the machine in order to rationalize the human. Advocate the human-machine system in order to amplify both. (Flach & Hoffman, 2012, p. 171)

When decision-making aids are designed to provide analysis or recommendations to aid in the decision-making process it is essential that the human decision-maker understands the factors considered, the weighting applied to the variables and outcomes, and the nature of the algorithm. The decision-maker must be supported in understanding the limitations and biases in the decision-making aid system's representation of the environment. A failure to account for these considerations can result in the system being underutilized due to a lack of trust, or potentially even worse, misused and over-trusted due a poor understanding of the system's capabilities and limitations.

b. Addressing Appropriate Levels of Trust and Familiarity

A JCS must be designed to ensure the operator trusts a decision-making aid enough to use it without placing too much faith in its recommendations. The operator must be able to understand the situations where the decision-making aid can and should be leveraged, understand how the aid is conducting its analysis, and understand what variables are left out of the algorithms altogether. This level of understanding is essential for the decision-maker to understand what variables he is responsible for evaluating and how to weigh those variables against the analysis provided by the decision-making aid.

To some extent this trust depends on training evolutions which provide declarative knowledge regarding the decision-making aid's algorithms and experience in leveraging the aid in cognitively complex environments. These issues are also heavily dependent on a decision-making aid interface which is intuitive and clearly demonstrates the system's recommendations as well as the limitations of the algorithms guiding those recommendations. Before determining how to develop the appropriate level of trust, however, the designer must determine the acceptable limits of system's sensitivity and response bias. In concert with calibrating the sensitivity of the technological system, the designer must determine how to calibrate the end user through interface design, work processes/heuristics, and training programs in order to mitigate under-trust, over-trust, and decision-making biases.

Determining what constitutes an "appropriate" level of trust in a decision-making aid depends on a consideration of the system's sensitivity and response bias. The designer must leverage system detection theory (SDT) to balance the design of the interface, data sources/sensors, and algorithms to increase likelihood of hits and correct rejections, while minimizing false positives and misses. In conjunction with end users and policy makers, they must identify the degree to which rates of false positives, referred to as "automation false alarms" or the "cry-wolf effect" with automated systems, are acceptable relative to rates of "misses" (Proctor & Van Zandt, 2018; Wickens et al., 2016).

This will be directly influenced by the nature of the decision-making tasks as well as the performance metrics set for the decision-making aid and the JCS as a whole. This analysis, while necessary for the design of the decision-making aid's algorithms and information source selection, will also guide the design of the user interface, work processes, and training program in order to develop an appropriate level of human trust in the system based on the situation.

The calibration of the technology must be conducted while simultaneously considering the risks of misuse by the human operator. If the operator is unable to recognize, and critically evaluate, the variables that he or she is responsible for evaluating relative to the machine, then they will likely misuse the decision-making aid. Over-trust can manifest as human-automation dependence, complacency, and/or automation bias (Wickens et al., 2016).

The cyber threat to military operations is a constant consideration in combat operations center and other such battle space awareness training. Personnel must remain capable of operating without the computerized C2 systems which support operations. One concern regarding decision-making aids is that the decision-maker may become dependent on the aid for decision-making and lose the ability to conduct independent analysis and decision-making. This can be mitigated by routinely challenging the decision-maker to train without the decision-making aid and designing the system so that it reinforces the skills necessary for performance when the system is unavailable. Another concern is the development of complacency, where the decision-maker fails to provide adequate oversight of the aid due to a high level of trust in the automated system.

These issues can result in what is known as "out-of-the-loop-unfamiliarity" (OOTLUF), where the decision-maker is unable to maintain adequate situational awareness because they are not actively involved in the sort of routine decision-making and information gathering which has been automated through a decision-making aid (i.e., the generation effect) (Wickens et al., 2016). This is related to the limitations of human vigilance, as it is difficult for a human to maintain vigilance for extended periods of time when a task involves monitoring a system with infrequently occurring events. Considering these issues leads to what may at first seem to be a counter-intuitive design recommendation. Rather than maximizing the automation of tasks, the designer should work with the end user to determine the appropriate level of operator workload to maintain desirable levels of vigilance and situational awareness. This should include an analysis of the operating environment, the cues relevant for supporting decision-making, and other C2 systems employed by the operator.

The operator's degree of trust in a decision-making aid also relies on the known, or perceived, rates of failure of the aid in the operating environment. Studies conducted in simulated environments have demonstrated that operator trust in an automated decision-making aid can be manipulated through the presentation of system reliability data and may also be impacted by the way such data is presented (Hollands & Neyedli, 2011). Perceived system reliability levels can also result from user experience in employing the system, though cognitive biases may skew the perception of reliability. The designer should leverage these studies to design the system to support an appropriate level of trust and design synthetic training environments which can support intuitive understanding of system reliability relative to the situation.

A related issue with decision-making aids is designing the system to present information in a way that mitigates decision-making bias. This includes designing the system to mitigate the many types of human decision-making biases which may come from the environment or their previous experience. The decision-maker can also be subject to automation bias, where they place more weight in the recommendations of the decision-making aid than are warranted by the limitations of the system and environment. These biases can be mitigated through user experience and interface design. In situations where the decision-making aid is conducting parallel analysis, for example, several biases (e.g., representativeness, availability) may be mitigated by having the user enter their analysis or weighting of variables into the interface before being presented the parallel analysis conducted by the decision-making aid. The designer should consult the study of heuristics and biases to find best practices commonly used to support human teams in limiting effects such as group think.

There are many additional design considerations which must be considered which are more fundamental to human factors. These considerations have been thoroughly explored in the study of human factors and human-computer interaction, but it is important that designers stay abreast of current research as continuously evolving technology (e.g., virtual reality, augmented reality, screen resolution improvements) may present new challenges and opportunities for system design.

c. Additional Human-Computer Interaction Considerations

In addition to ensuring the interface presents useful cues to support recognition primed decision-making and avoids biasing decision-making process, the user interface design should be guided by sensory, anthropometric, and microcognitive capabilities and limitations of the user. These include factors such as the ability discriminate signals (sensory and cognitive limitations), change blindness, and working memory. These factors are not always as clear cut as they are sometime presented. The limitations of short term memory, for example, are not as clear cut as the "7 plus or minus 2" chunks, explanation which is often presented (Flach & Hoffman, 2012; G. A. Miller, 1956). The size and composition of such 'chunks' are relative to the expertise of the operator. Here again, the design must anticipate the influence of human experiences and training design on human limitations.

It is important that the microergonomic considerations be considered relative to the macroergonomic environment. Microcognitive factors in particular must be considered relative to the macrocognitive functions (e.g., sensemaking) which support the user's decision-making. As has been seen in the study of dual-task performance for vehicle drivers using "in-vehicle information systems" (IVIS), macrocognitive functions influence how operator's interact with their environment, and must be considered in the evaluation of technology interfaces (Lee, 2010). Just as "drivers are not the passive recipients of IVIS and roadway demands," decision-makers are not passive recipients of the decision-making aid interface (Lee, 2010, p. 101). The design of the decision-making aid must consider the microcognitive factors relative to macrocognitive functions, and vice versa.

2. Holistic Development and Refinement of a JCS

It is often stated in the military that a plan never survives first contact with the enemy. This expression serves as a reminder that mission success requires a continuous reevaluation and refinement of operations. The design and implementation of a JCS is no different. System designers must anticipate changes in the operating environment and plan for numerous evaluations and refinements following system fielding.

One important challenge with the development of information technology such as decision-making aids is anticipating their impact on the environment they are designed to support. This challenge commonly manifests in the form of new technology which is fielded without adequately defined work processes and performance models (Klein & Ralls, 1997). This is not due to a lack of motivation on the part of the system developers, as significant investments are generally made in both system design and training for implementation. Instead, this issue can be attributed to what the CSE community refers to as the “envisioned world problem.”

The envisioned world problem describes the challenge of anticipating how the introduction of a new technology (or work process, organizational structure, or training) will impact the dynamics of the JCS as a whole. Introduction of a decision-making aid can provide a decision-maker valuable new insights or analytical support, but it can also introduce the sorts of trust, workload, and decision-making biases discussed previously. Furthermore, decision-making aids should be employed in accordance with their capabilities, not in accordance with prior work processes.

Recall Gary Kasparov’s assertion that a “weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process” (Kasparov, 2017, p. 246). This statement highlights the importance of strengthening and aligning all three cognitive triad elements in a JCS. Recognition of the importance of the process in human-machine teaming is also mirrored in the focus of many companies on the development of “‘IA’, or intelligence amplification, to use information technology as a tool to enhance human decisions instead of replacing them with autonomous AI systems” (Kasparov, 2017, p. 248; Daugherty & Wilson, 2018). Developing an understanding of the current work domain and forecasting future performance requires the observation of existing work domain and the use of synthetic environments which can stress and evaluate the JCS in current and anticipated future work environments.

3. JCS Evaluation

Design and development of a JCS requires an understanding of the decision-making task(s) and environment. This includes identification of environmental constraints, best practices, and performance measures used for evaluation of the decision-maker and the JCS as a whole. Following clarification of the desired improvements in JCS performance (e.g., identifying optimal solutions or satisficing with lower decision-maker workload or time requirements), this analysis can be conducted through CTA and cognitive work analysis (CWA) methods for existing operating environments. Insights gleaned from existing work environments can inform the study of a JCS in anticipated future work environments, including the design of simulations for exploring JCS performance in anticipated future work environments.

CTA methods should be used to develop an understanding of the cognitively complex facets of the tasks associated with the domain (Hoffman & Militello, 2009). These methods can support the identification of decision-making best practices, limitations, and the environmental cues relevant to supporting cognitively complex tasks, including the recognition primed decision-making process (Hoffman & Militello, 2009; Kirwan & Ainsworth, 1992). CTA methods can also support the development of a corpus of challenging training scenarios with high variability. As will be discussed later, such scenarios can support both training simulations to develop decision maker experience in leveraging the new aid and simulations which identify the limitations in the system. The latter identification of JCS "edges" of performance is critical for supporting the continuous improvement of all elements of the JCS (Potter & Rousseau, 2010).

While CTA methods focus on the nature of the tasks within the work environment, CWA focuses on the constraints which define the work environment itself (Hoffman & Militello, 2009). Analysis of the decision-making environment can include the use of CWA methods for exploring the affordances of the environment (the Abstraction Hierarchy) and the "strategies and heuristics that experts might use to satisfy their functional goals" (the Decision Ladder) (Flach & Voorhorst, 2020a, p. 293). These frameworks support analysis which goes beyond the conventional, dualistic "analytic versus intuitive" approach to decision-making, to consider the role of the ecology in the decision-making environment (Bisantz & Vicente, 1994; Flach & Voorhorst, 2020a). The Abstraction Hierarchy supports a differentiation of the decision-making environment's "'deep structure' (e.g., more abstract constraints related to intentions and laws of the [domain]) from 'surface structure' (e.g., more concrete constraints associated with physical properties of a situation)" (Flach & Voorhorst, 2020a, p. 186). Rasmussen's Decision Ladder supports an appreciation of how mental models of a domain, a decision-makers understanding of the 'deep structure' (and recognition of patterns), can support intuitive decision-making and heuristics.

The influence of the ecological perspective can also be leveraged to develop an appreciation for the ways different decision-making theories may apply to the decision-making situations encountered by the JCS. Analyzing dimensions of the decision-making environment can provide insight into the types of decision-making problems which are likely to be faced (Flach & Voorhorst, 2020b). This in turn can guide the identification of decision-making approaches to reinforce with the design of the decision-making aid, process design, and/or user training (e.g., normative, recognition primed decision-making, cybernetic decision-making).

This analysis, which is critical for the development of an IS in a JCS, can also inform the development and evaluation of user training and process design. An understanding of appropriate decision-making approaches for various situations can inform the selection of training events and the cultivation of appropriate scenarios. Wargaming environments, which are useful in exploring decision-making processes in novel domains, may be better designed given an understanding of where and how different decision-making approaches are leveraged by humans and machines of the JCS.

a. CSE Methods for JCS Evaluation

CSE methods for simulating complex work environments in order to evaluate joint cognitive systems support the continuous evaluation and refinement of JCS components. The Decision Centered Testing (DCT) methodology, for example, is designed to support a rigorous evaluation of the JCS by stressing it across all possible "facets of complexity" in the operating environment (Patterson et al., 2010; Potter & Rousseau, 2010). This methodology supports the identification of JCS "edges," which may be improved upon through modifications to any, or all, elements of the JCS.

A failure to take such a holistic, iterative approach to JCS design and evaluation can result in the delivery of an information system which actually makes staff coordination and decision-making "more difficult, more drawn out, and more susceptible to error" (Salmon et al., 2010, p. 129). By conducting scenario-based simulations throughout the design and development of cognitive systems like decision support systems, system designers and end users can ensure effective feedback is generated to guide the development of both the decision-making aid and the associated work processes and training programs. Following fielding of the cognitive system, DCT evolutions can be integrated into training environments to support the continued development and evaluation of the decision-makers, work processes, and technology.

Anticipating how changes in one element will impact the other cognitive triad elements of a JCS can be extremely difficult. The challenge of anticipating how changes to one element of the JCS affects the other two has been compared to modeling challenges in physics, where “modeling the motion of interacting bodies in space becomes significantly less tractable when a third body is introduced” (Flach, 2013, p. 3). The difficulty of forecasting the evolution of a JCS is compounded by the challenge of forecasting changes in the future work domain itself, as the work domain is often continuously evolving due to external factors and advances in adjacent technologies. Woods and Hollnagel (2006) identify “three classes of research methods” (p. 44) for the evaluation of JCSs in context, listed below.

1. Natural History methods (in situ)
2. Experiments-in-the-field: Staged or Scaled Worlds
3. Spartan lab experiments: Experimenter-created artificial tasks (Woods & Hollnagel, 2006, p. 44)

These environments afford different levels of control and authenticity (Woods & Hollnagel, 2006), illustrated in Figure 7, to help researchers investigate a JCS in past, current, and envisioned future work domains.

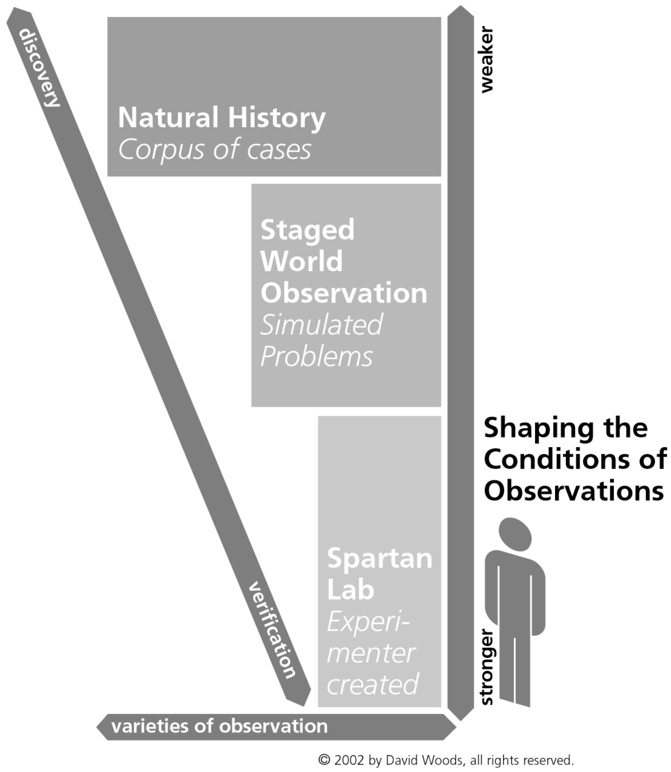


Figure 7. Research Method Environments for JCS Evaluation. Source: Woods and Hollnagel (2006).

Natural history methods can include observation of real-world operations (unlikely for military and emergency management operations) or the use of critical incident methods for analysis of previous events (Woods & Hollnagel, 2006). These methods have high authenticity, as they address real-world JCSs in their natural environments, and provide insights into the constraints of the past or current work environment. Natural history methods afford little to no control to researcher, however, for adjusting the work domain for exploration of specific scenarios or adjusting other artifacts of the JCS. With spartan labs (also known as micro-worlds (M. J. Miller & Feigh, 2019)), researchers sacrifice authenticity for high levels of control over what's simulated. Spartan labs abstract away many of the work domain dynamics in order to focus on specific relationships among cognitive triad elements.

b. Staged World Simulations for JCS Evaluation

Staged worlds provide a middle ground, facilitating the study of current JCSs dynamics and the evaluation of JCS modifications in context (M. J. Miller & Feigh, 2019).The context can be either the current work environment or some envisioned future work environment. Staged worlds are "simulations of work contexts that focus on specific situations or problems that practitioners may encounter and that preserve key interrelationships in the cognitive systems triad" (Sanderson, 2018, p. 108). The simulated work domain dynamics may be representations of anticipated changes to the JCS itself, perhaps forecasting the coevolution, or they can represent changes which are anticipated due to advances in adjacent technologies, force structure, or operating environments.

> The effectiveness of a staged world rests in how effectively the essential properties of the cognitive systems triad are preserved in the experiences created – experiences that emerge from the relationship between people, technology, and work. A staged world can create situations that might arise only very seldom in naturalistic observation, while still preserving key properties of the work domain that create an authentic, immersive experience for practitioners. As a result, a staged world is an effective and efficient means of investigating cognitive work. (Smith & Hoffman, 2018, p. 108)

Staged worlds are particularly important for the continuous coevolution of JCSs which are exercised infrequently in the real world. For military and emergency management organizations, staged worlds can provide rare insight into the potential performance of infrequently exercised JCSs. This can be seen in the Adaptive Architectures in Command and Control (A2C2) research project, initiated by the Office of Naval Research (ONR) in 1995 to explore how military organizations may need to adapt for the information age; see Figure 8. Under the A2C2 project, different organizational structures and processes were modeled and simulated through a combination of "computational and humans-in-the-loop experimentation" (Smith & Hoffman, 2018, p. 131), and even "'organization-in-the-loop' experimentation" (Smith & Hoffman, 2018, p. 125), to improve the design of organizational structures and processes with the networks and information systems of the information age.

> The goal of the A2C2 program was to develop and test ways to optimize fit or congruence among (1) the human agents involved in a task; (2) the mission, tasks and work processes they were conducting; (3) the technical capabilities they were utilizing to do that work; and (4) the informational, sociocultural, and organizational structures that the sociotechnical organization is embedded in. (Smith & Hoffman, 2018, p. 121)

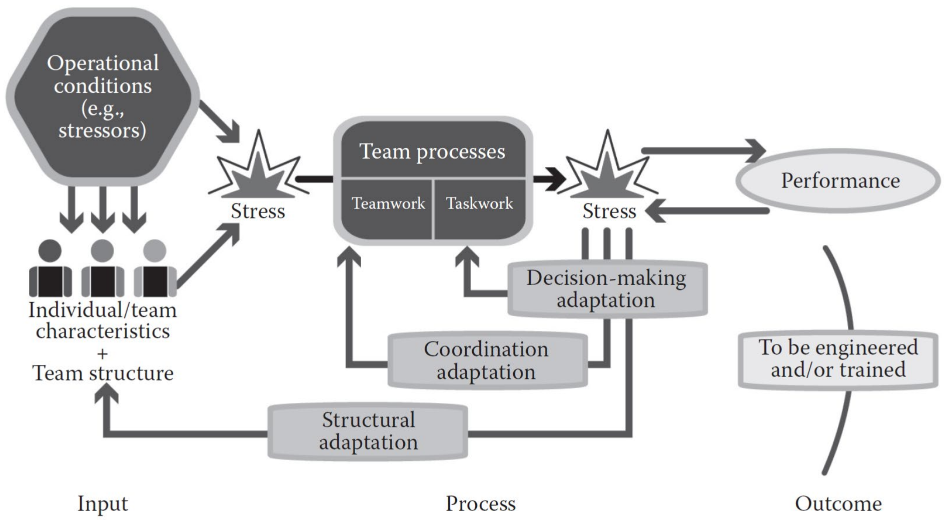


Figure 8. Feedback Loops in High Performing Organizations. Source: Smith and Hoffman (2018).

Simulation-supported environments used for studying JCSs may also be leveraged to address the challenge of "perceptual relearning of dynamic integral transmodal cue configurations" (Hoffman & Fiore, 2012, p. 210). This issue, which is associated with the HCC "moving target rule," describes the challenge many experts face with maintaining expertise within naturalistic decision-making domains where the information systems are frequently replaced or modified. This applies to environments where experts make decisions based on "dynamic information defined over sets of integral cues that are transmodal (they exist over different data types)" (Hoffman & Fiore, 2012, p. 210). Such environments can include business management, weather forecasting, and combat or emergency management operations centers. "One way of thinking about it is that it takes the notion of pattern recognition to entirely new levels. Another way of thinking about it is that it's a cautionary tale about the rather nebulous yet popular notion of 'information fusion'" (Hoffman & Fiore, 2012, p. 210).

In these environments, the decision-makers and staff actions are often informed by numerous complex information systems which evolve over time. The time required for adapting expert skills to new technology may be mitigated by providing simulation-supported, scenario-based training environments with adequate functional fidelity and a range of challenging scenarios with high variability which is representative of the challenges which may be encountered in the actual operating environment.

> The Moving Target Rule: The sociotechnical workplace is constantly changing, and constant change in environmental constraints (such as technologies in the workplace) might entail constant change in cognitive constraints (the work to be accomplished), even if the domain constraints remain constant. (Hoffman & Fiore, 2012, p. 210)

C. SIM-IS ENVIRONMENT DESIGN AND DEVELOPMENT

Significant progress has been made in achieving technical and semantic interoperability between simulations and C2 systems, and the ideal of conceptual interoperability across distributed simulations is well established. No method currently exists, however, for designing sim-IS information exchange to simulate the way an information system is populated by an organization's existing, or anticipated, sociotechnical system. Designing sim-IS environments to simulate BP-IS integration is necessary for developing an understanding of the capabilities and limitations of their information systems in context. This is important for both JCS evaluation and the development of trainee mental models.

For the dynamics of sociotechnical systems to be adequately represented in simulation-supported training and wargaming environments, sim-IS environment conceptual models (CMs) must be developed to support the design of sim-IS information exchange. Sim-IS environment designers and developers must also evaluate the composability of simulation and IS conceptual models, regardless of the sim-IS information exchange method employed (e.g., RPA, puckster). ***The modeling and simulation (M&S) community's existing simulation environment design and***

development frameworks are insufficient for the design of sim-IS environments. Sim-IS environment designers and developers need guidelines for the development of sim-IS environment CMs which use BP-IS models as referents. This framework, or overlay to an existing framework (e.g., DSEEP), must guide the design of sim-IS environment CMs and their use in the development of sim-IS environments which simulate the dynamics of BP-IS integration in target business contexts.

1. Conceptual Modeling for Distributed Simulation Environments

The design of simulations and multi-simulation environments depends on the use of conceptual models which capture the entities, relationships, and attributes of the referent to be represented. Simulation conceptual models are “a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model” (Robinson, 2008, p. 14). They are either abstract mental models, existing only in the minds of the simulation designers, or explicit models capturing a subset of the referent in a formalized way to support a common understanding for the design and modernization of simulations and distributed simulation environments.

For individual simulations, explicit simulation conceptual models enable users to understand what aspects of the referent are represented in the simulation and how. They facilitate an evaluation of a simulation’s suitability for specific modeling questions and training requirements and facilitate the modernization of simulations as the referent and modeling questions evolve. For distributed simulation environments consisting of multiple simulations, conceptual models facilitate an evaluation of simulation composability for a given set of modeling questions or training requirements.

The creation of explicit conceptual models depends on the use of ontologies. Here “ontology” refers to “a formal specification of a conceptualization, which fulfills the requirements for a conceptual model” (Tolk & Miller, 2011, p. 133). Ontologies used for M&S are either methodological or referential (Hofmann et al., 2011; McGinnis et al., 2011). Methodological ontologies define a formal language for conceptual modeling (e.g., unified modeling language [UML], systems modeling language [SysML]). A referential (or domain) ontology supports a shared understanding of the referent and "describes structural and behavioral aspects of the target domain" (McGinnis et al., 2011, p. 191).

While the potential for ontologies to support explicit conceptual modeling has been well established (McGinnis et al., 2011; Tolk & Miller, 2011; Turnitsa et al., 2010), the fundamental nature of ontologies, especially the balancing of consistency and completeness, precludes the establishment of a universal conceptual modeling standard. Consistency refers to the ability of all elements of an ontology to align with one another. Completeness refers to the ability of an ontology to reflect all aspects of a domain. This poses a problem for domain ontologies, as completeness in some domains results in inconsistencies. Such inconsistencies are not a problem for reference models, where completeness is prioritized over consistency, but conceptual models must be consistent in order to yield working simulations (Jones, 2015). For a referential ontology to be complete enough to support every domain, it would not be consistent enough to support conceptual modeling. This precludes the development of any universal referential ontology. No universal methodological ontology can be established either, due to unique modeling requirements of different domains.

These fundamental limitations of ontologies preclude the development of universal domain or methodological ontologies for conceptual modeling (Jones, 2015). Unlike the M&S interoperability standards maintained by the Simulation Interoperability Standards Organization (SISO) (e.g., Distributed Interactive Simulation [DIS], High Level Architecture [HLA]), conceptual modeling is guided by best practices and supported by domain-specific ontologies (both methodological and referential) (Jones, 2015; Tolk, 2017). While some conceptual modeling approaches have been developed using UML and SysML, these are often intractable for non-engineers (Morse & Drake, 2021). The Multi-Viewpoint Conceptual Modeling (MVCM) approach attempts to resolve this by providing a conceptual model format and development process which are accessible to various stakeholders (e.g., simulation engineers, training developers, senior leaders). Higher level guidelines for conceptual modeling and the role of conceptual models in the design and development of distributed simulation environments are provided by the Distributed Simulation Engineering and Execution Process (DSEEP) depicted in Figure 9.

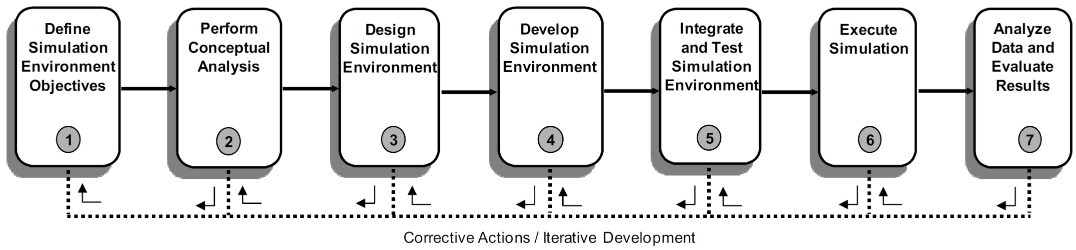


Figure 9. Distributed Simulation Engineering and Execution Process (DSEEP) Top-Level View. Source: *IEEE Std 1730* (2011) © 2011 IEEE.

DSEEP's step 2, perform conceptual analysis, describes the development of a conceptual model which defines the elements of the referent to be represented in the simulation. Throughout the DSEEP steps, this conceptual model is "transformed from a general representation of the real-world domain to a more specific articulation of the capabilities of the simulation environment as constrained by the member applications of the simulation environment and available resources" (*IEEE Std 1730*, 2011, p. 15).

The conceptual model guides the design of the simulation environment (DSEEP step 3), including the selection of member applications/simulations, allocation of functions or entities to applications, and design of member applications. This includes evaluation of the suitability of simulations for simulating assigned entities and functions for the target environment conceptual model. The conceptual model also guides the development of the simulation environment (DSEEP step 4), including the design of information exchange between member applications through the simulation data exchange model (SDEM).

While much of a conceptual model's functions and processes may be ascribed to individual member simulations, for some environments, elements of the conceptual model are associated with the exchange of information between member applications. The design of this information exchange depends on the member application sending the information, the member application receiving the information, and any middleware needed to modify the nature of the information exchange to align with the conceptual model. This is especially important for sim-IS environments, where BP-IS integration is simulated through the member applications and the information exchange mechanism which connects them. ***DSEEP does not currently provide guidance for sim-IS environment developers***

regarding the design of sim-IS environments to address the design of information exchange between simulations and information systems. DSEEP also does not provide guidance regarding any unique requirements for the development of a sim-IS data exchange model compared to conventional SDEMs. The identification of sim-IS CM and data exchange model requires an appreciation of how conceptual modeling is leveraged in the development of information systems and the modeling of integrated business processes.

2. Related Research: Extending DSEEP for Sim-C2 Interoperability

For sim-IS environments to effectively simulate sociotechnical systems an explicit methodology is needed to guide the evaluation of composability (between simulations and ISs) and the design and development of information exchange between the component simulation(s) and IS. Current methodologies for the design and development of distributed simulation environments, and the integration of information systems in those environments, do not address these considerations. The primary tool for the design of such environments is the DSEEP. The DSEEP provides high level guidance, and overlays have been developed which provide additional guidance for issues like integration of simulations with different architectures (e.g., DSEEP Multi-Architecture Overlay [DMAO]).

While DSEEP identifies the necessity for conceptual analysis/modeling as prerequisite for the development of distributed simulation environment conceptual models, this framework does not address the unique challenges associated with conceptual modeling of sim-IS environments. DSEEP does not address the design and development of information exchange between simulations and ISs to simulate the dynamics of BP-IS integration for a particular sim-IS environment, as illustrated in Figure 10. Given the growing importance of sim-IS interoperability for staff training environments, the M&S community would benefit from a DSEEP overlay which addresses the development of conceptual models and information exchange data models for such sim-IS environments.

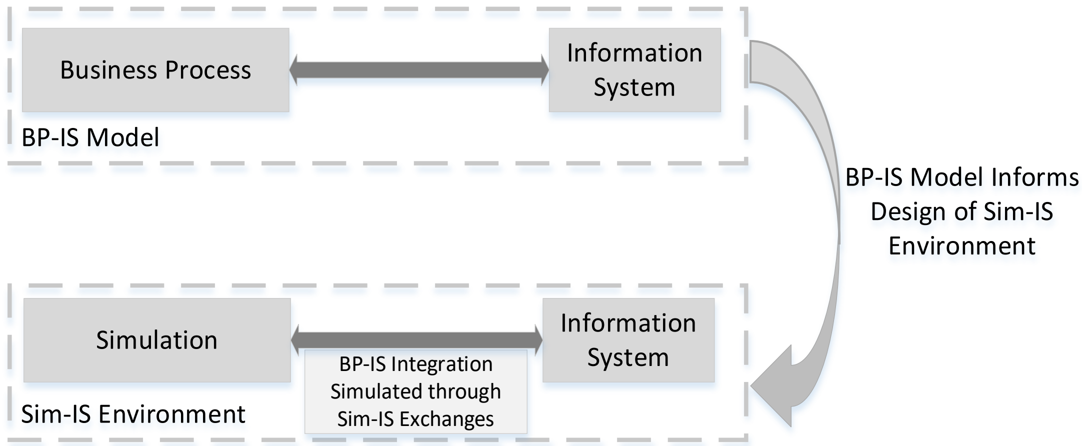


Figure 10. BP-IS Model as the Referent for Sim-IS Environment Design.

Sim-IS environment conceptual models are necessary to support the design of sim-IS information exchange and evaluate the composability of simulations and information systems relative to the integrated business processes intended to be supported by the sim-IS environment. Like simulations, information systems are built using abstractions (conceptual models) of a referent. Unlike simulations, which are intended to emulate a referent without interacting with it, information systems interact with the business processes they support. This results in a recursive element of IS conceptual models, with IS CMs reflecting the IS structure/function as well as its anticipated role in the business processes it will support. This poses a challenge for IS conceptual modeling, as the introduction of the IS will necessarily change the business process it is intended to support.

This IS CM challenge is one reason information system development approaches have migrated toward Agile and DevOps methods. The recursive nature of IS CMs must also be addressed in sim-IS environment CMs, where the conceptual model must capture the structure/function of the live IS as well as the nature of its interoperation with business processes simulated by the simulation system and any sim-IS information exchange middleware. Developers of sim-IS environments would benefit from guidance for the design of sim-IS environments and the continuous coevolution of sim-IS environment CMs and BP-IS models.

Just as the DMAO and the (emerging) Verification, Validation, and Accreditation (VV&A) DSEEP overlays support the application of DSEEP to more specific distributed simulation environments and challenges, a DSEEP overlay is needed which supports the unique challenges associated with sim-IS environment design and development. An overlay has previously been proposed which would address the technical and semantic challenges of sim-C2 interoperability. North Atlantic Treaty Organization (NATO) Modeling and Simulation Group (MSG) 085 proposed the development of a C2SIM DSEEP overlay which would provide "recommended practices for applying DSEEP to the development and execution of distributed simulation environments that involve one or more C2 systems used to command and control autonomous simulated entities" (Heffner et al., 2014, pp. 2–8).

The NATO MSG-085 Technical Activity proposed a C2SIM DSEEP Overlay which would provide guidance for the development of C2SIM federations (NATO Science and Technology Organization, 2015b). A C2SIM federation is defined as "a simulation environment that contains at least one C2 system, and that uses a C2SIM data exchange model as the SDEM" (Heffner et al., 2014, p. 9). Issues recommended for inclusion in the C2SIM DSEEP overlay would include the avoidance of information overload (of C2 systems), filtering of simulation information for that required by the C2 system, and routing of traffic between select units' systems.

While issues such as "simulation information filtering" and "ground truth and perceived truth" are related to broader sim-IS information exchange challenges addressed in this research, the proposed C2SIM DSEEP Overlay was intended to address these issues relative to the technical routing of information between operational C2 systems (Heffner et al., 2014; Standardisation for C2-Simulation Interoperation, 2015). The proposed overlay was not intended to address the design of information exchange between simulations and information systems to simulate the integration of information systems in business processes. Instead, the proposed C2SIM DSEEP overlay would primarily support the development and execution of C2SIM federations by addressing the technical and semantic interoperability of simulations and C2 systems. Conceptual modeling considerations necessary for ensuring sim-IS information exchange adequately simulates the integration of the information system in the target business processes were not considered.

While achieving technical and semantic interoperability is important for the design and development of sim-IS environments, those levels of interoperability are not sufficient for addressing the first two layers of sim-IS interoperability, especially for HitL ISs. ***To address these layers of sim-IS interoperability, a methodology is needed which supports the design and development of sim-IS environments which represent the dynamics of information exchange across sociotechnical systems (e.g., information delay, human error).***

3. Conceptual Modeling of Information Systems and Sociotechnical Systems

Some researchers have suggested that the field of M&S is unique in its reliance on conceptual models for capturing the referent. Tolk et al. (2012) asserted that "while other interoperability domains connect real things and can refer to the same real-world referents, M&S interoperability connects conceptualizations, and we have to understand what the participating systems concepts look like in order to operate together" (Tolk et al., 2012, p. 12). Turnitsa et al. (2010) claimed that IS ontologies are "used to describe an assumed objectively observable system" (Turnitsa et al., 2010, p. 650). This is an incomplete representation of the role of conceptual models in the design and development of information systems and the coevolution of information systems with other elements of associated sociotechnical systems.

Information modeling has long had an important role in supporting the design and maintenance of information systems. Some of the earliest models supported the conceptualization and design of information systems in the form of entity-relationship models and semantic models for databases in the 1960s and 70's (Mylopoulos, 1992, 1998). With the introduction of requirements engineering in the 1970s, models were needed which supported IS design relative to the user requirements and envisioned work environment. These models came to be known as conceptual models (Borgida et al., 1984).

With increasing complexity across the various components of information systems and requirements engineering, conceptual modeling was intended to provide a high-level medium for synchronizing the efforts of IS designers, users, and managers. Mylopoulos (1992) defined conceptual modeling as “the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication” (Mylopoulos, 1992, p. 2). This definition leaves a great deal of room for interpretation. Some researchers have focused on ensuring conceptual models provide formal descriptions by prioritizing the completeness and rigor of modeling languages and methods. Others have prioritized understanding and communication, asserting that “the fundamental characteristic of the new level of system description is that it is closer to the human conceptualization of a problem domain” (Brodie et al., 1984, p. vi). As information systems and associated work environments grew increasingly complex, modeling languages grew more specialized, making it increasingly difficult to balance completeness and rigor against comprehensibility. At the same time, information systems researchers recognized that the cyclical nature of work artifact and task design requires continuous reengineering of both tasks and information systems (or work artifacts more generally).

As tasks and artifacts coevolve, as shown by the task-artifact cycle in Figure 11, conceptual models provide a means of documenting and informing the coevolution of information systems and their associated work environments. This dynamic relationship also provides a moving target for designers of information systems and work environments, such that conceptual models serve as the designer’s conceptualization of how a system may be employed as opposed to a representation of an “objectively observable system.” Unlike engineering models which directly associate to programming code and database structures, information system conceptual models situate information systems in conceptualizations of fluid sociotechnical systems.

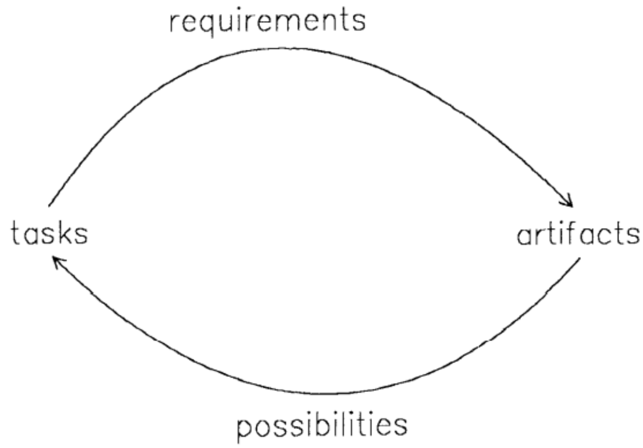


Figure 11. The Task-Artifact Cycle. Source: Carroll and Rosson (1992).

Conceptual models provide guidance for the design of information system, business process, and user training, regarding the current or anticipated structure and dynamics of a target element and/or environment. Conceptual models are used to support the design and maintenance of information systems by associating an IS design with an abstraction of the existing, or anticipated, work environment. Conceptual modeling is also leveraged to support the continuous alignment of coevolving business processes and information systems. At a more granular level, abstract models of tasks and associated user interfaces guide the design of user interfaces and target mental models to support user selection and training design.

Despite the original intent of conceptual modeling to support a holistic view of information systems and the nature of their employment, the increasing specialization of computer sciences domains (e.g., information systems, user interface design, artificial intelligence) resulted in a variety of specialized conceptual modeling languages and methodologies (Kaschek, 2008). Today, various modeling languages exist, tailored for supporting different domains, types of systems, and levels of abstraction. While much research has focused on the development of these specialized conceptual modeling approaches within specific communities, and extension of existing languages, several research communities have attempted to integrated conceptual modeling approaches for more holistic design and management of sociotechnical systems and joint cognitive systems. Building on training, CSE, and process modeling concepts, Erhart and Bigbee (1999) assert the value of integrating “multiple models, including user profiles, decision and functional task models, organizational models (goal hierarchies, control structures, processes, and functions), hardware, software & communication architectures, information models (data structures, and information flow), human-computer interaction models (information presentation and interaction models, and collaboration models)” (Ehrhart & Bigbee, 1999, p. 15).

A multimodel approach, illustrated in Figure 12, is now leveraged to varying degrees across different research communities for the design and analysis of sociotechnical systems (Pennock & Gaffney, 2018). Multiview modeling was one of the early information system modeling approaches intended to support a more holistic design of information systems in context (Krogstie, 2012). It was developed as a “framework for information systems development that would take account of the complex world of people and organizations as well as information and communication technologies” (Avison et al., 1998, p. 1). While the multiview modeling approach provides flexibility to use heterogeneous modeling languages to better describe disparate elements of a sociotechnical system, it also presents designers an additional challenge in aligning the models.

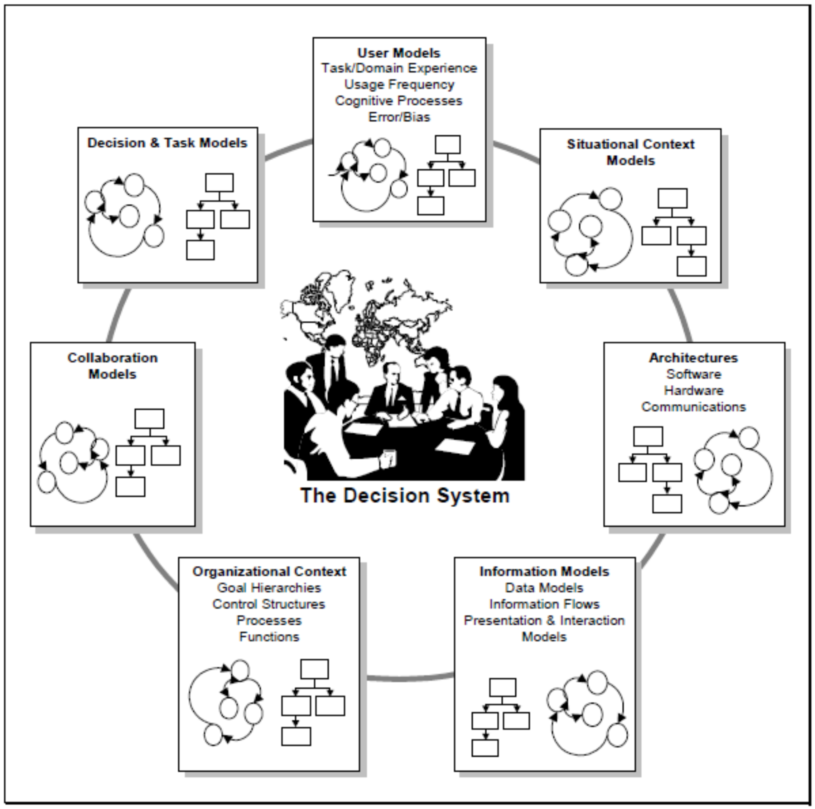


Figure 12. Multiple Models for Defining an Organizational Decision System. Source: Ehrhart and Bigbee (1999).

The use of multiple modeling methods allows designers to leverage the relative strengths of different modeling languages for different components of the ISs and broader STSs they are attempting to represent. There are several factors to consider when selecting modeling languages. Within any individual domain, selecting the right modeling language requires consideration of a modeling language's "domain appropriateness, which refers to truthfulness of the language to the domain [and] comprehensibility appropriateness, which refers to the pragmatic efficiency of the language to support communication, understanding and reasoning in the domain" (Guizzardi et al., 2005, p. 691). When considering the use of multiple modeling languages across multiple domains, a designer must also consider the extent to which different modeling languages can be integrated, to the extent their project requires integration of different views. Understanding how to approach these considerations requires accounting for the practical impact of the relationship between conceptual modeling languages and referent ontologies discussed earlier. This can be best discussed by contrasting two general approaches to conceptual modeling languages: general-purpose modeling languages and domain-specific modeling languages.

General-purpose modeling languages (e.g., UML) are designed to have broad applicability. The ontologies with which they are associated, foundational ontologies, are broad by definition and "should be valid across domains and contain very abstract concepts only" (Gonzalez-Perez, 2017, p. 6). Domain-specific modeling languages are often extensions of general-purpose modeling languages but afford greater precision within a particular domain defined by a domain ontology, as illustrated in Figure 13. The domain ontologies with which domain specific modeling languages are associated are often extensions of foundational ontologies, built to provide a "formal and explicit specification of a domain conceptualization" (Guizzardi et al., 2002, p. 4). As modeling languages and domain ontologies are extended for specific domains, their precision increases for the given domain and their generalizability to other domains generally decreases.

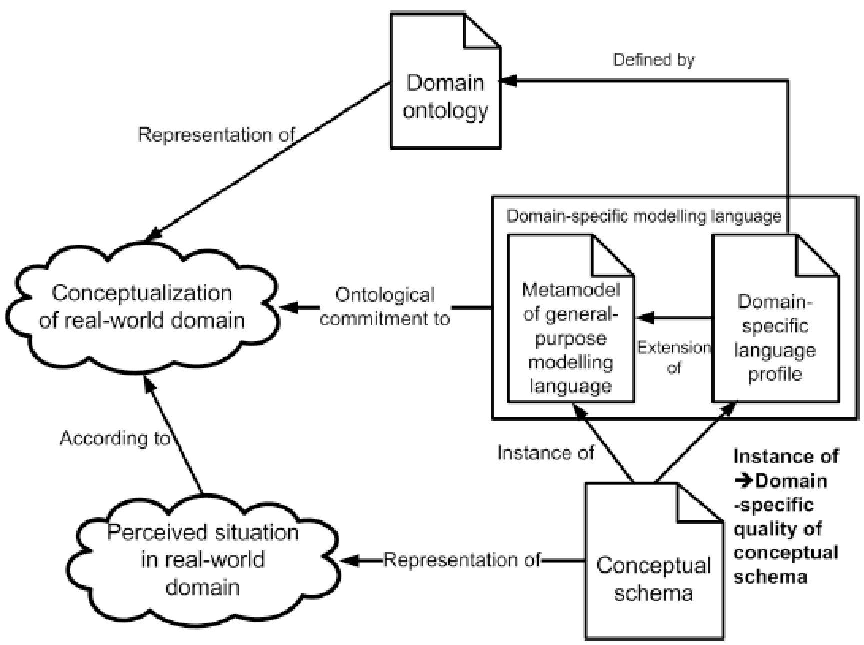


Figure 13. Domain Ontology-based Conceptual Modeling Approach. Source: Gailly and Poels (2010).

"One of the main success factors for a domain specific language is its ability to provide to its users a set of modeling primitives that correspond to relevant domain abstractions" (Guizzardi et al., 2002, p. 4). McGinnis et al. (2011) categorize three ways ontologies are commonly used to support simulation development: "ontology for simulation modeling, ontology for domain modeling, and a hybrid approach based on ontology integration" (McGinnis et al., 2011, p. 2). Focusing on the third category, they present an approach for defining domain-ontologies and associated domain-specific modeling languages through Systems Modeling Language (SysML), demonstrating how the resulting domain-specific languages can support the automated translation of conceptual models to computational simulation models. This use of SysML for both ontology definition and domain modeling has been lauded as providing a "complete integration between defining and using the ontology [which] helps in faster development and leads to implementable Domain Specific Languages (DSL)" (Jain et al., 2015, p. 3).

Modelers can determine the domain appropriateness of a modeling language by "comparing the level of homomorphism between a concrete representation of the worldview underlying the language (captured in a metamodel of the language)," with the domain ontology for the target domain (Guizzardi et al., 2005, p. 3). Homomorphism refers to the alignment between different representations of a domain or phenomenon. A homomorphism can be said to exist between a metamodel and a given domain ontology, for example, if each element of the metamodel can be mapped to an equivalent element of the domain ontology. The "level of homomorphism" refers to the degree to which the metamodel elements are mapped to unique elements of the ontology (injective mapping) and the degree to which all ontology elements are mapped to a metamodel element (surjective mapping) (Gurr, 1998); refer to Figure 14.

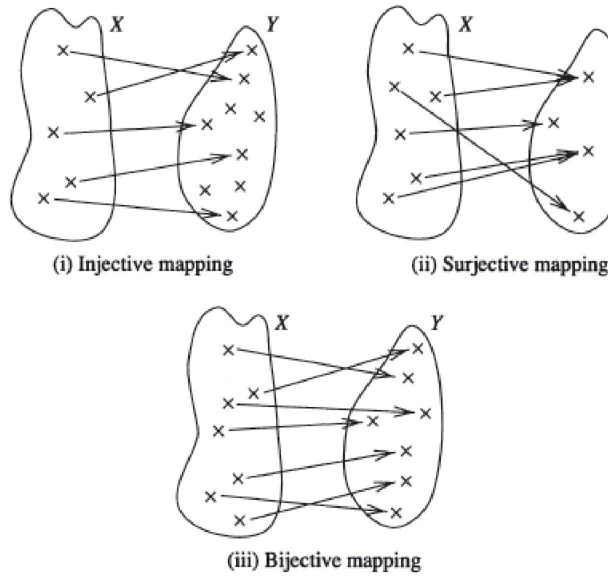


Figure 14. Levels of Homomorphism Between Representations of a Domain or Phenomenon (X and Y). Source: Gurr (1998).

When a homomorphism includes both injective and surjective mapping, it is said to include bijective morphism and is known as an isomorphism. Homomorphism can occur in both directions. Models can be built with different modeling languages but share a common domain ontology, or vice versa (Guizzardi et al., 2002). An isomorphism, however, can only exist between a single pair of models, or between a given metamodel and domain ontology pair. Guizzardi et al. (2005) present a "framework for "evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages" (p. 14) by evaluating the degree of homomorphism between the metamodel of a given modeling language and the domain ontology for the target domain of study. If a metamodel and domain ontology are completely aligned, such that any conceptualization built within one would result in a guaranteed interpretation within the other, the metamodel and domain ontology would be considered isomorphic.

While isomorphism is the highest level of alignment between a metamodel and an associated ontology, such a high level of alignment is not always necessary or even desirable. Often, different domain specific modeling languages are desirable for use in modeling different aspects, or views, of a shared domain. Guizzardu et al. (2002) present a scenario where two different modeling languages (represented as $\mathscr{L}_1$ and $\mathscr{L}_2$ in Figure 15) are used to model different attributes of a shared domain conceptualization (CD), with "logical interpretations π_1 and π_2 that relate the formal semantics of $\mathscr{L}_1$ and $\mathscr{L}_2$ to a common domain ontology OD" (Guizzardi et al., 2002, p. 8). In such a scenario, a common domain ontology can serve to align the different modeling languages to support different but aligned views of a given domain.

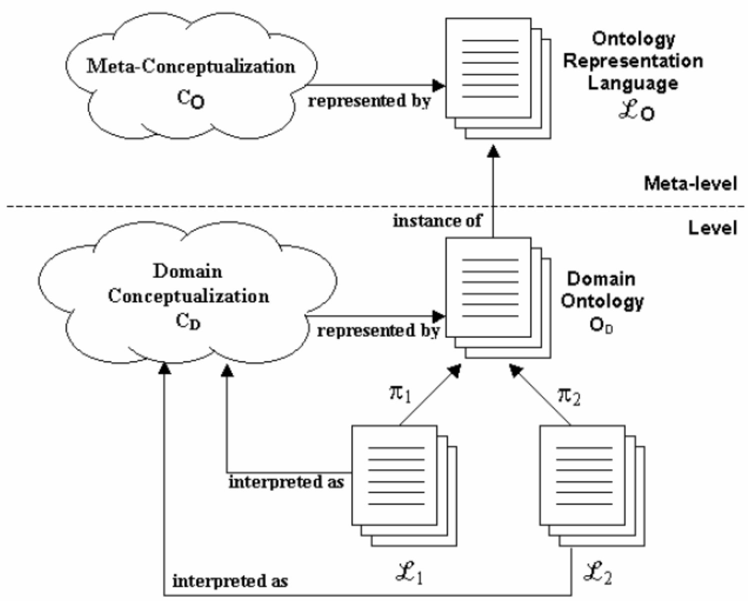


Figure 15. Aligning Domain Specific Modeling Languages ($\mathscr{L}_1$ and $\mathscr{L}_2$) Through a Shared Domain Ontology (OD). Source: Guizzardi et al. (2002).

Modelers also face situations where different domain specific modeling languages must be used to describe different domains of a single system, with different domain ontologies. In these instances, different approaches can be taken for the alignment of models, depending on the degree of rigor required in the mapping of the multiple views. With the emergence of model-based and model-driven engineering, tight integration of models is often necessary to support the automated transformation of abstract models into concrete models for the development of software, user interfaces, or support automated simulations. In these situations, ensuring consistency across the multiple models is a common challenge (Bork, 2015).

If domain specific modeling languages derive from a common general-purpose modeling language (e.g., UML extensions), the general-purpose language can support translation between domain specific models. Other approaches to modeling multiple views with tight integration include "transformation from one modeling language into another, matching models via their meta models, or concentrating on managing models of the same kind" (Fengel & Rebstock, 2010, p. 1). Integration of models using different modeling languages also requires alignment of their associated ontologies, to ensure semantic alignment. If the domain ontologies derive from a common foundational ontology, the foundational (or "upper") ontology can support alignment of common concepts.

> [Foundational] ontologies establish a structure to which domain ontologies can conform, by serving as a starting point to build new (domain) ontologies, as a reference for the comparison of different (domain) ontologies, and as a common framework for (domain) ontology harmonization and integration. (Gonzalez-Perez, 2017, p. 6)

In multi-view modeling, designers may also model different views separately, with no integration or alignment of the disparate models, described in Table 1. Examples of this "independent metamodel design" approach include the Zachman framework or the DOD Architecture Framework (DoDAF) (Bork & Alter, 2020). The DoDAF presents a general framework for the development and management of conceptual models for any type of system acquired or maintained by the DOD. The hierarchical structure of DoDAF provides separate levels, or "views," which are intended to provide different insights into the system and its intended application. The higher-level views illustrate how a system will fit into the operating environment it is intended to support and integrate with other systems. Lower-level views provide the detailed engineering and data standards information which is necessary for achieving system development, maintenance, and integration with other systems.

Table 1. Metamodel Design Options. Source: Bork and Alter (2020).

Design option	Strengths	Weaknesses
Integrated Metamodels	• Tight integration of all techniques • No syntactic inconsistencies • Extensions only affect one artifact	• Complex overarching metamodel • All changes need to be valid globally, i. e., for all techniques
Independent Metamodels	• Efficient development • No side-effects • Efficient extensions and update	• No syntactic coupling between different techniques • One technique at a time
Interlinked Metamodels	• Efficient development • Different techniques can be coupled, e. g., along a more complex modeling procedure	• Bidirectional transformations, effortful when new techniques are added • Side-effects (moderate)

"The three metamodel design options provide different means of realizing coherence on the semantic level and integration on the syntactical level. Method engineers need to decide which type of integration is most suitable for the method at hand. Independent metamodels are more flexible, whereas integrated metamodels allow specifying an integrated model which leads to more powerful analysis possibilities" (Bork & Alter, 2020, p. 11).

In recent years, several researchers in the field of enterprise modeling (which addresses the modeling of "large-scale sociotechnical systems" (Bruseberg, 2008, p. 3)) have argued that conceptual modeling of sociotechnical systems requires a relaxation of the rigor associated with the tight integration of modeling methods found in integrated metamodel design(Bork & Alter, 2020; Jasperson et al., 2005; Sandkuhl et al., 2018). Sandkuhl et al. (2018) advocate a softening of the "completeness, coherence and rigor requirements" of modeling (p. 72). The proposed relaxation, they argue, would make conceptual models more accessible to all stakeholders and responsive to evolving work systems.

Bork and Alter (2020) propose a work system modeling method (WSMM) which charts a path between fully integrated and independent metamodel designs, with interlinked metamodels envisioned as the primary design option to be used. The WSMM derives from the conceptualization of the "work system" as a "unit of analysis for thinking about systems in organizations" (Alter, 2013, p. 75, 2008; Jasperson et al., 2005). Like the CSE approach to joint cognitive systems, work systems theory "views the social and the technical as part of a single system" (Alter, 2013, p. 91). While cognitive systems engineering supports the coevolution of the work environment, tools, and personnel, more rigid business process models are rarely included in CSE research, as the community often focuses on naturalistic cognitive work. Work system theory (WST) focuses on the integration of business processes and information systems, with less attention to the mental models and naturalistic decision-making environments which are often central to CSE research.

The WSMM builds on the hybrid approach to multi-view modeling, which was developed to alleviate the problems of consistency management associated with integrated metamodels while still supporting the design of complex sociotechnical systems (Cicchetti et al., 2012a, 2012b). Another approach developed in response to the increasing complexity and intractability of conceptual models is the Multi-Viewpoint Conceptual Modeling (MVCM) method (Morse & Drake, 2021). The MVCM method similarly attempts to overcome the inaccessibility of the highly structured conceptual modeling methods by using a combination of scenario, visual graphics, and tables to present a simulation environment representation that can be understood by non-engineers. The emphasis this approach places on model accessibility resonates with the original intent of conceptual models to support a "human conceptualization of a problem domain" and "enhance communication between system designers, domain experts and, ultimately, system end-users" (Brodie et al., 1984, p. vi).

Almost twenty years have passed since Wand and Weber asserted that conceptual modeling "plays an increasingly important role in activities like business process reengineering and documentation of best-practice data and process models in enterprise resource planning systems" (Wand & Weber, 2002, p. 1). If sim-IS environments are to represent the dynamics of complex sociotechnical systems, then sim-IS conceptual models must include models of integrated business processes as well as a conceptualization of how they will be represented in sim-IS environments. This requires an understanding of what conceptual modeling approaches, and languages, are best suited for capturing the variety of aspects of sim-IS environments. This section provides an overview of conceptual modeling approaches for representation of integrated business processes, IS user interfaces, and user mental models.

a. Conceptual Models of Integrated Business Processes

Business processes are activities which are executed by one or more functional units of an organization in support of the organization's mission (Venkatesan, 2018). Documentation of business processes in the form of business process models serves many purposes. For large, distributed organizations, standardized business processes facilitate communication and coordination by providing a common conceptualization (to some extent) of the associated business process for disparate units of an organization. Explicit business process models can also support analysis for process improvements (e.g., continuous process improvement projects), guide personnel training design, and support the design of tools (including information systems) used in support of work. Modern business processes can be viewed from two orthogonal perspectives: the IT and business views (Dijkman et al., 2008; Ulmer et al., 2013). Conceptual modeling for information systems and business processes faces the same challenge of balancing completeness and consistency discussed in the context of M&S conceptual modeling (Mineau et al., 2000). No one language can simultaneously be precise and general enough to support the modeling of all concepts and relationships required for modeling business processes and ISs in all domains. Instead, a variety of modeling languages exist which support business process modeling and/or IS modeling to varying degrees.

Information systems, especially those which human operators use directly, are designed based on an imperfect abstraction (conceptual model) of a fluid sociotechnical system. Just as M&S researchers recognize the importance of pursuing higher levels of simulation interoperability, IS and business process management researchers have increasingly pursued business process-IT alignment (BITA) (Badr et al., 2016; Clark & Jones, 1999; Tolk, 2006; Venkatesan, 2018). Achieving BITA has proven especially challenging for cross-functional enterprise resource planning (ERP) systems, and many organizations experience poor IS launches due to an inadequate design of BP-IS integration (Al-Mashari et al., 2003). As information systems proliferated, and the BP-IS integration became more complex and cross-functional, business process redesign in support of BITA has been increasingly recognized as essential for the effective employment of new information systems (Al-Mashari et al., 2003). **Two major obstacles to achieving BITA**

are the integration of business process and information systems models and the need to continuously refine both models as integrated business processes coevolve(Millet et al., 2009).

All information systems, whether they are developed in house or acquired from external vendors, are built upon "implicit business models" developed from the IT perspective (Butler et al., 2000; Millet et al., 2009, p. 402). Butler et al. (2000) argue that the implicit business models associated with each IS should be made explicit in IS conceptual models, which should also capture the structure/function of the IS itself and the nature of BP-IS integration (i.e., how the IS will interact with the envisioned business processes). A primary challenge in the development of such explicit documentation of the implicit business model of an IS is translating between different business process and information system modeling languages. Though the implicit business models may be informed by explicit business process models built by business practitioners in the associated domain, there is often a misalignment in the models resulting from orthogonal business and IT perspectives, commonly referred to as the "business-IT gap" (Ulmer et al., 2013).

> Both process descriptions [business process models and workflow specifications] cover the same matter of interest: the involved tasks and their order. The close relation between the two descriptions suggests deriving one from the other by changing the level of abstraction, e.g., enhancing existing business process descriptions such that they can be used as inputs for a WFMS [workflow management system]…So far, there is no methodologically well-founded process model that bridges the gap between business process- and workflow modeling. (Dehnert & van der Aalst, 2004, p. 290)

While business process modeling and workflow modeling are closely related, and the terms are often used interchangeably, they provide unique perspectives on information systems from the business process management and information system sides respectively (Pütz & Sinz, 2010). Mili et al., 2004 describe business process models as including "both automated (computerized) activities and processes" while IS models "[focus] on the computerized part" (p. 5). These differences in scope and perspective necessitate the use of different modeling languages to represent dynamics from the business and IT perspectives respectively.

> We should make it clear that even a description of a business process and the corresponding workflow specification both refer to the same set of activities and their ordering are simply expressed at different levels of abstraction with different viewpoints…A business process description is made by domain experts…The objective of a business process description is to provide a basis for communication…A workflow specification, in contrast, is made by IT experts...Whereas it is sufficient for a business process description to cover the set of desired process executions, a workflow specification also determines how these executions are achieved. (Garcia, 2010, p. 7)

The integration of business process and IS models faces the same problems of consistency and completeness that prevent the generation of a single modeling language for business processes and information systems. There are multiple taxonomies of modeling languages and information systems. In their "Taxonomy of Business Modeling and Information System Modeling Techniques," Giaglis (2001) identifies the tendency of modeling languages to support either business for IT concepts, despite the well-established, recursive relationship between business process and IS design. Krogstie et al. (2008) defines five usage categories for process models:

1. Human sense-making and communication to make sense of aspects of an enterprise and to support communication among different stakeholders. Sense-making models are used within an activity in order to make sense of something in an ad-hoc manner and will usually not be maintained afterwards.

2. Computer-assisted analysis to gain knowledge about the enterprise through simulation or deduction.

3. Business Process Management, following up the adherence of the work process to standards and regulations. Here the model is meant to act as part of a corporate memory meant to exist as a reference point over time.

4. Model deployment and activation to integrate the model in an information system. Deployment can be manual, automatic (in automated workflow systems), or interactive.

5. Using the model as a context for a system development project, without being directly implemented (as it is in category 4). (Krogstie et al., 2008, p. 309)

Mili et al. (2010) present four categories for the variety of modelling languages used in modeling integrated business processes:

1. traditional process modeling languages (e.g., integrated computer-aided manufacturing definition [IDEF], Petri Nets, Event Process Chains),

2. object-oriented languages (e.g., UML),

3. dynamic process modeling languages (e.g., workflow modeling languages, business process execution language [BPEL]),

4. and process integration languages (RosettaNet, choreography description language [CDL]).

Traditional process modeling languages are commonly used in business process design and reengineering. Their emphasis on being accessible to humans makes them accessible to most stakeholders, but also limits their precision and ability to support software considerations. Object-oriented languages, such as UML are "geared more towards representing the solution (software) domain rather than the problem (business) domain" (Mili et al., 2010, p. 10). UML was originally designed to support modeling for software design. While UML was extended (and updated with UML 2) to support representation of some business process modeling concepts issues, it is generally considered ill-equipped for business process modeling compared to traditional process modeling languages (Butler et al., 2000; Giaglis, 2001; Mili et al., 2010).

Dynamic process modeling languages support executable business process models for workflow management systems or process aware information systems. These languages are more precise than traditional modeling languages, prioritizing being machine readable and executable over accessibility to human stakeholders. While Mili et al. (2010) include business process modeling notation (BPMN) in this classification, BPMN is in some ways more closely aligned with the traditional process modeling languages. BPMN is designed for human use rather than being machine readable and executable, and is commonly used as an accessible business process modeling language in industry. For example, Stein et al. (2009) present BPMN as a business modeling language in their research addressing the alignment of business process and workflow models, using BPEL instead as an example of a workflow (or dynamic process modeling) language. Process integration languages are generally designed to support the modeling of electronic "business-to-business" (B2B) transactions. They "typically focus on the mechanics of the integration in terms of abstract, technology independent, programming interfaces and data exchange formats" (Mili et al., 2010, p. 12).

Object-oriented modeling has been the subject of significant efforts for BP-IS modeling, with many researchers and business practitioners attempting to employ object-oriented modeling languages like UML to model both business processes and ISs. Isoda (2001) identifies risks associated with such use of object-oriented modeling languages for both business and IS domains, deriving from different perspectives required for the different modeling "usages" seen below.

> The purpose of object-oriented real-world modeling, or, in other words, the usage of class diagrams made by real-world modeling, can be classified into three cases, namely
>
> Usage 1. To facilitate understanding of a problem in the real world without any intention of developing an application based on the diagram.
>
> Usage 2. To develop an application that simulates the real world.
>
> Usage 3. To develop an application that automates business in the real world. (Isoda, 2001, p. 155)

He asserts a fundamental difference in how object-oriented languages are used for modeling in support of usages 1 and 2, deemed "genuine real-world modeling," compared to their employment in support of usage 3, which requires the use of an additional object-oriented modeling method deemed "pseudo real-world modeling." The difference between genuine real-world modeling and pseudo real-world modeling scenarios can be partly ascribed to the difference between the “real-worlds” with which the respective models are associated. While usages 1 and 2 are based on the assumption of a single, objective reference “world,” seen in Figure 16, the use of object-oriented modeling languages for the development of ISs (usage 3) requires a differentiation between the “original” and “automated” (or envisioned) world. This concept, illustrated in Figure 17, is somewhat similar to the envisioned world problem discussed earlier. Isoda defines the process for genuine real-world modeling in support of usages 1 and 2, and a mixed pseudo and real-world modeling method for usage 3.

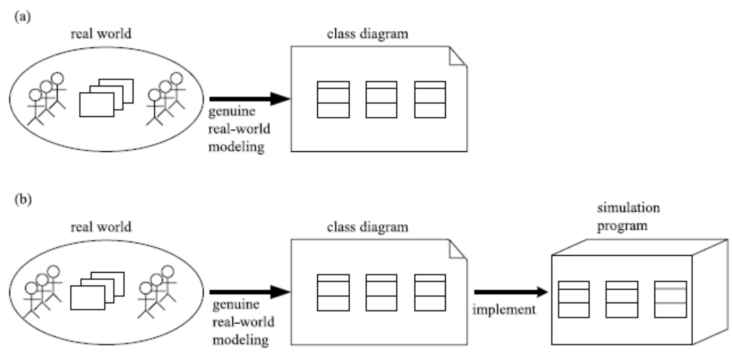


Figure 16. Genuine Real-World Modeling: (a) Usage 1; (b) Usage 2. Source: Isoda (2001).

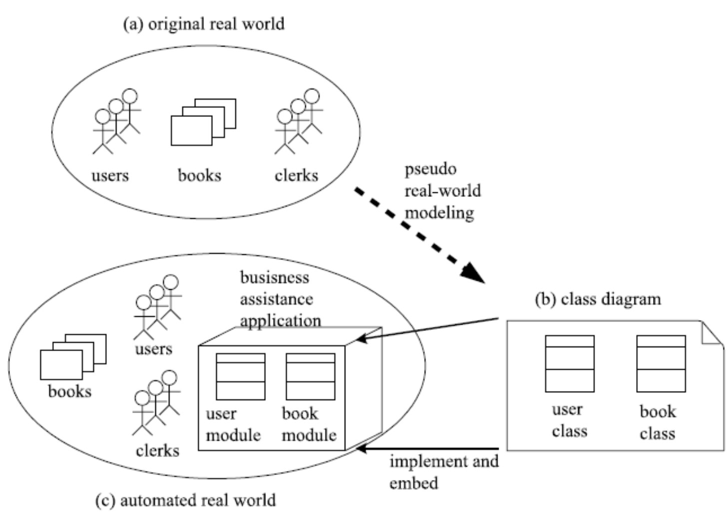


Figure 17. Pseudo Real-World Modeling: Usage 3. Source: Isoda (2001).

Building on the work of Isoda (2001), Mili et al. (2010) assert that "business process modeling requires real-world modeling of the business and its processes, while software modeling requires pseudo real-world modeling" (p. 10). In a usage 3 instance, an information system may be represented as a "black box" in the automated real-world model of an IS and the pseudo real-world model captures the internal functioning of the IS. Of note, the entity classes (for example user and book) are not equivalent between the two models. While the entities included in a "real-world" model of a business process and operating environment are intended to represent the actual entities, the entity classes in a pseudo real-world model of an IS instead represent the information about those entities which are relevant to the IS. The literature is less clear on the challenge of modeling scenarios which may fall between usages 2 and 3. In such scenarios, the "real world" to be simulated may require the combination of real-world and pseudo real-world modeling, due to the unclear role of given ISs in the environment.

Since Mili et al. (2010) provided their classification of modeling languages for integrated business processes, the gap between business process and IS models has remained a persistent challenge. During this time, service-oriented architecture IS approaches have proliferated, resulting in the development of yet another IS-focused modeling language, service oriented architecture modeling language [SoaML], and additional research exploring methods for integrating business process models (e.g., BPMN) with the IS-centric SoaML models (Elvesæter et al., 2010; Leshob et al., 2019).

> The lack of synchronisation and absence of semantic equivalence between models prevent us from obtaining what is called as an "intermodal" consistency. This is the famous Business-IT gap found in the literature (Peppard and Ward 1999, Grembergen 2004, Stein et al. 2009), the gap between business and IT domains. (Ulmer et al., 2013)

While these different modeling languages are tailored for modeling business processes and information systems for their respective domains and levels of abstraction, the variety of general-purpose and domain-specific business process and IS modeling languages poses a challenge for conceptual model integration in support of BITA (Butler et al., 2000; Ehrhart & Bigbee, 1999; Fengel & Rebstock, 2010; Trætteberg, 1999; Venkatesan & Sundaramurthy, 2018). Approaches for employing heterogeneous models have been categorized as consisting of "(1) transformation of process models, (2) meta2-modeling and ontologies, and (3) standardization of methods and tools"(Frerichs et al., 2021, p. 1568). There is, of course, some overlap between these approaches. Transformation of models, for example, generally requires a comparison of meta-models to ensure semantic and alignment (Ulmer et al., 2013). Each approaches includes its own strengths and weaknesses, and no one approach has proven universally appropriate for integration of business process and information system models.

One approach for the modeling of integrate business processes is to use multiple modeling languages, leveraging different modeling languages for their respective domains in a problem space. This can include coupling an object-oriented modeling language (UML) for IS modeling and a traditional process modeling language (IDEF3) for work modeling (Butler et al., 1999). Butler et al. (2000) suggest that the problem of model integration may be resolved, at least in part, by establishing an explicit relationship between the class diagram of the object-oriented modeling language and the business model built using a process modeling language. In such an instance the class diagram of a general-purpose object-oriented modeling language would essentially serve as an explicit domain ontology to align the IS and business process models for a particular business domain. This approach has matured in recent years with growing demand for tightly integrated models and with automation of alignment of modeling language and domain language semantics supported by development of semantic web technologies (Fengel, 2013; Fengel et al., 2014).

Fengel and Rebstock (2010) present a more thorough approach for aligning disparate models by leveraging semantic-web technologies to support "linking models in different modeling languages as well as different model types" through a "bridge ontology" (Fengel & Rebstock, 2010, p. 1). This bridge ontology is called the Unifying Modeling Concepts Ontology (UMCO), and is designed to integrate Modeling Concepts Ontologies (MCO) developed for each of the different modeling languages to be integrated (Fengel, 2013). Figures 18 and 19 illustrate how the alignment of modeling language semantics, captured in respective MCOs, via a UMCO (Figure 18) facilitates the semantic alignment of models built using different modeling languages (Figure 19). Following modeling language alignment, domain languages would also need to be aligned, as illustrated in Figure 20, with “[domain] ontology matching and information linguistics methods” supporting the determination of “semantic correspondence that describes equivalence to a certain degree and thereby content similarity” (Fengel, 2013, p. 101).

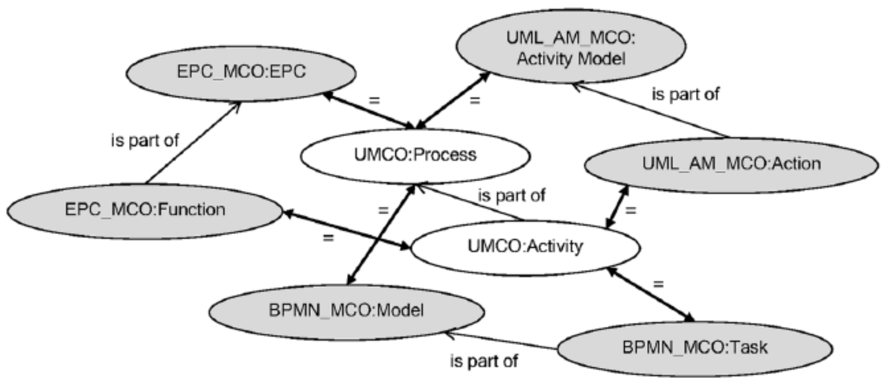


Figure 18. Extract of the UMCO Showing Semantic Alignment of Modeling Languages. Source: Fengel (2013).

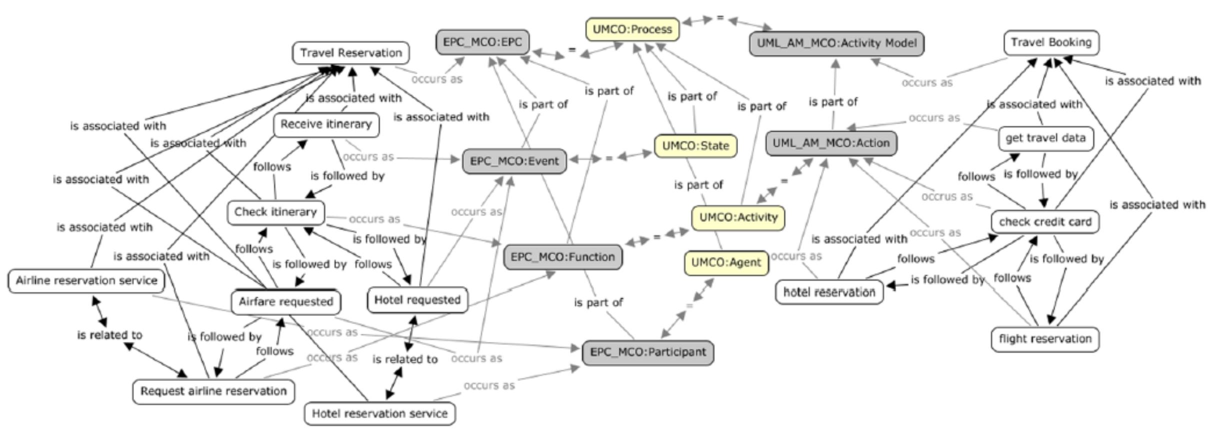


Figure 19. Aligning Event-Driven Process Chain (EPC) and UML Modeling Language Semantics via UMCO. Source: Fengel (2013).

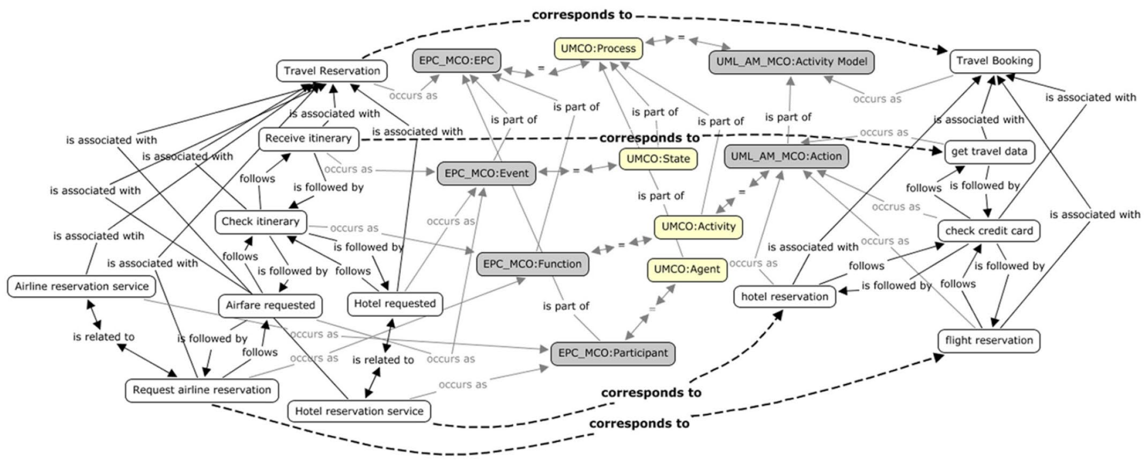


Figure 20. Linking Domain Semantics of EPC and UML Models. Source: Fengel (2013).

Another approach for modeling integrated business processes is to extend one modeling language to address both business process and information system modeling considerations. Extensive work in this approach has resulted in extensions and updates to UML for business process modeling as well as BPMN extensions for representing information system considerations. These approaches often also tailor to extensions of general-purpose modeling languages for representing the business process or IS considerations of a particular domain. An example of one such domain-specific integrated business process modeling solution is provided by Millet et al. (2009), who extend the supply chain operations reference (SCOR) model to represent information exchange concepts required for supporting alignment of business processes and information systems in the business and supply chain management domain. This requires extension of the SCOR model to support representation of information objects within the business/supply chain management domain.

As previously mentioned, tight integration of modeling views is not always necessary or desired. Enterprise architecture frameworks can support the alignment of business processes and information systems, with more loosely aligned models. Enterprise architectures have been described as a "strategic approach to facilitate business-IT alignment" (Alaeddini et al., 2017; Venkatesan & Sundaramurthy, 2018, p. 2), as they provide a common framework for modeling and comparing the various levels of systems developed across an organization. Under the DOD Architecture Framework (DoDAF), for example, business processes which information systems are intended to support could be captured in the "operational view" level. Some researchers have even argued that these DoDAF "views" could serve as models for simulation designers (Van Den Berg & Lutz, 2015). Unfortunately, such explicit conceptual models of information systems and their envisioned role in integrated business processes are not generated or maintained for many DOD systems. This is the case for some Marine Corps logistics information systems (e.g., CLC2S), where associated business processes and BP-IS integration are not standardized.

The integration of business process and IS models are made even more complicated by the evolution of both business processes and ISs over time in response to changes in each other and their respective domains. Figure 21 shows how the development of a business process model and an associated dynamic process model (or orchestration model) over time can result in misalignment, as the process and IS change over time (Stein et al., 2009). In this example, business domain personnel first build a business process model (BPM), and then, following some refinement, partner with IT personnel to transform the BPM into an "abstract orchestration" model to support the design of IS(s) intended to support the business process(es). The IT personnel refine their model independent of the business personnel and both the BPM and IS models evolve over time before an effort to resynchronize them at a later time. A notable limitation of this representation is that it only shows the IS model (orchestration) as being influenced by the BPM, and does not show the influence of the IS on BPM redesign.

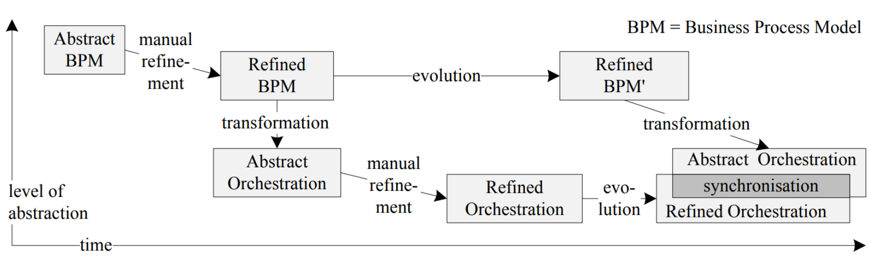


Figure 21. Coevolution of Business Process Models and IS Models (Orchestration Models). Source: Stein et al. (2009).

This is the **second obstacle** presented by Millet et al. (2009), which requires a process which supports continuous evaluation and re-design of information systems (e.g., DevOps) in support of a **continuous coevolution of business processes and information systems**. The SCOR model-based approach to BITA, proposed by Millet et al. (2009) and seen in Figure 22, addresses this obstacle by continuously iterating between process redesign relative to the IS and IS re-design relative to adjustments in the business process. The business process redesign (BPR) for process alignment (PA) with ISs includes refinement of best practices (BP) relative to "software functionality (features) that enable best practices" (Millet et al., 2009, p. 398). Throughout such a process of continuous coevolution, it is important to maintain conceptual models for both the IS and associated business process to support such iterative coevolution in pursuit of enhanced BITA (Butler et al., 2000).

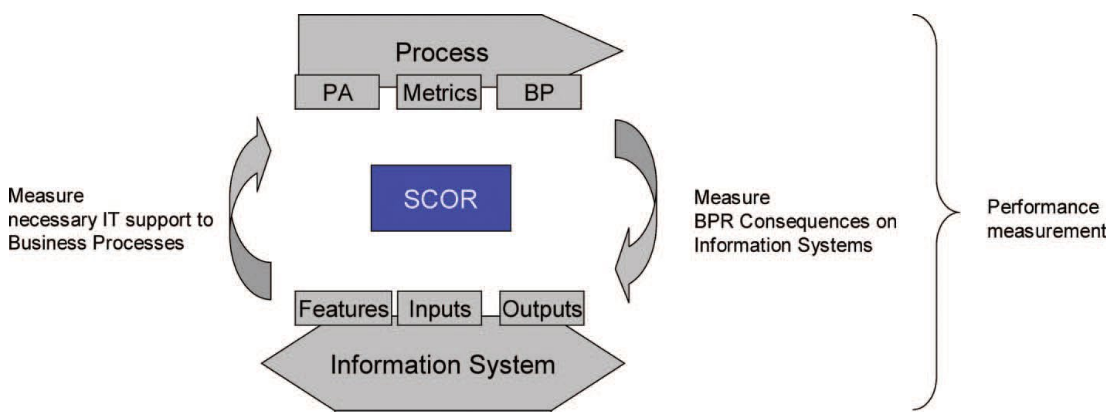


Figure 22. The SCOR Model Based Approach for the Alignment of Information Systems and Business Processes. Source: Millet et al. (2009).

The challenge of continuous BP-IS alignment also extends to user tasks and system interfaces, necessitating the integration of models addressing different levels of abstraction (Sousa et al., 2009; Trætteberg & Krogstie, 2008). The value of integrating the various levels and perspectives of integrated business processes (e.g., workflow modeling, task modeling, user interface modeling), has long been appreciated within the information systems community (Butler et al., 2000; Trætteberg, 1999). This can present an obstacle as different modeling languages may be used to model the different levels of abstraction. One way this has been addressed is through the extension of existing modeling languages, or development of new modeling languages, to support both business process and task modeling (Pontico et al., 2006; Trætteberg & Krogstie, 2008). Just as different modeling languages are often required for modeling business processes and ISs, however, it is also generally recognized that no single modeling language is suitable for all levels of abstraction across all possible domains (Bork & Alter, 2020; Van Wyk & Heimdahl, 2009). Another approach therefore includes the transformation of different modeling languages for the integration of business process, workflow, and task models (Garcia, 2010; Murzek, 2008; Trætteberg, 1999).

> Workflow modeling languages are often used for describing organizational and group work, while task analysis and task modeling are used for describing and formalizing individual work. Both essentially describe the same domain, but at different levels. (Garcia, 2010, p. 37)

While both business process and workflow modeling include representation of tasks, they are generally focused on the ordering of tasks relative to processes (workflow and process models) and identification of resources required for task execution (process models) (Garcia, 2010). Task models provide "a more precise description of each step of [a] workflow model" (Pontico et al., 2006, p. 5), with a user-centric perspective which differs from the system and organization views of business process and workflow models. These differences reflect the different origins of the modeling approaches, with process and workflow modeling originating in work organization domain while task modeling originates from the human factors domain (Pontico et al., 2006). Task models are particularly important for identifying what actions and mental models are required of personnel managing or operating within a sociotechnical system (Puerta-Melguizo et al., 2002; van der Veer & Puerta Melguizo, 2002; West & Nagy, 2007). Task models are also necessary for supporting the design of graphic user interfaces and ensuring user interface design is aligned with the design of the associated integrated business processes (Trætteberg & Krogstie, 2008).

b. Modeling of User Interfaces Relative to Integrated Business Processes

While much research has been conducted regarding the alignment of business processes and information systems, such efforts have often stopped short of supporting the modeling of IS user interfaces and their relationship to BP-IS alignment (Sousa et al., 2009). Modeling integrated business processes down to the user interface level requires a decomposition of business process or workflow models into task models and alignment of task models with user interface models. Common languages for modeling the business processes and ISs of integrated business processes (e.g., BPMN, BPEL, UML) are insufficiently equipped for modeling user tasks (Trætteberg & Krogstie, 2008). This limits their utility in supporting software user interface design and capturing the effects of business process/IS changes on user tasks associated with user interfaces (Kristiansen & Trætteberg, 2007; Pontico et al., 2006; Sousa et al., 2009). "[Business processes] could not be directly associated to [user interfaces] because they represent the business context and some of their characteristics make them a limited representation for [user interface] design" (Sousa et al., 2010, p. 3).

Task models are necessary for mapping business process and user interface models. While business processes describe the "structured set of activities, performed by organizations' stakeholders," task models describe how exactly those activities are executed (Sousa et al., 2010, p. 2). Task models, which often have a hierarchical structure, can be decomposed to different levels of abstraction, supporting task association with the granular components of user interfaces as described in UI models, see Figure 23. A variety of task analysis and modeling approaches have been developed over the past decades to support such fields as cognitive psychology, task planning, software engineering, and ethnography (Bowen et al., 2021; Limbourg & Vanderdonckt, 2004). Task modeling approaches are one element of task analysis methods, which may include task models, stepwise approaches to describing tasks, and/or software tools for developing task models (Limbourg & Vanderdonckt, 2004). "Despite the fact that various specific task notations exist, they are mainly structured around two concepts: task decomposition (often represented as a hierarchy) and task flow (for showing the order in which tasks are executed)" (Martinie et al., 2011, p. 590).

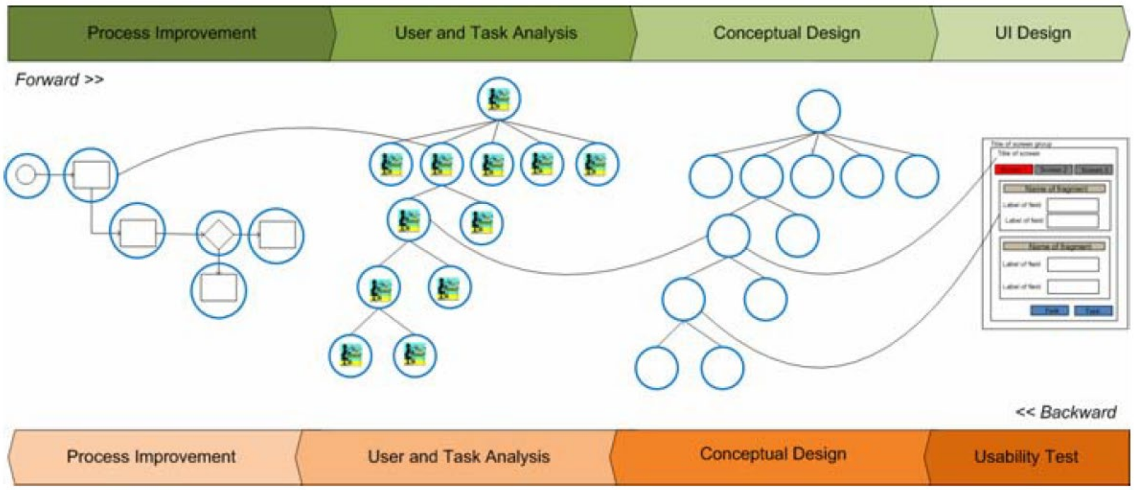


Figure 23. Integration of Business Process, Task, and User Interface Models. Source: Sousa et al. (2010).

Hierarchical approaches to task analysis are particularly well suited to capturing the dynamics of complex sociotechnical systems (Salmon et al., 2020). The most common of these is hierarchical task analysis (HTA). HTA supports the representation of cognitive and physical tasks in a work environment by facilitating a decomposition of work systems "into a hierarchy of goal, subordinate goals, operations and plans" (Salmon et al., 2020, p. 356). Because of its emphasis on decomposing systems and flexibility in supporting a wide range of work systems, HTA is often coupled with various other task analysis techniques (Kirwan & Ainsworth, 1992). Another hierarchical approach to task analysis, which is commonly used for decomposition of tasks associated with human-computer interaction, is the goals, operators, methods, and selection (GOMS) approach. With GOMS, the specification of goals and subordinate goals is followed by description of each subordinate goal "in terms of operations required to achieve it" (Kirwan & Ainsworth, 1992, p. 395). "HTA, GOMS etc., are often used interchangeably for naming the analysis/modeling approach and the resulting task model. Similarly simplifying, a HTA/GOMS/…model does not refer to a task model per se but to a type of task models" (Bowen et al., 2021, p. 4).

Task modeling approaches have different strengths including degrees of expressiveness, usability, and interoperability with other modeling languages. Different task models are also designed to support different task analysis/modeling objectives such as capturing "potential usability problems" (e.g., HTA), modeling human performance (e.g., GOMS), or "providing a detailed task model describing task hierarchy, objects used, and knowledge structures" (e.g., Concur Task Trees [CTT]) (Limbourg & Vanderdonckt, 2004, p. 136). Traditional task modeling approaches (e.g., HTA, GOMS, CTT) provide a "comprehensive notation to support a variety of task and action types, hierarchical decomposition and support for device and dialog model integration" (Bowen et al., 2021, p. 2). However, they can also be overly complex and time intensive depending on the size and importance of the task model(s) for a given problem. Some user interface designers and researchers have foregone the use of such traditional task modeling approaches by extending common modeling languages (e.g., BPMN, BPEL, UML) for modeling UI-associated tasks for a particular domain (Auer et al., 2009; Kovacevic, 1999; Lee et al., 2008; Trætteberg & Krogstie, 2008). This approach can present problems for model reuse as ad hoc notations and narrowly engineered languages can be less generalizable than the more comprehensive languages, requiring rework as designers "'reinvent' task modelling notations each time" (Bowen et al., 2021, p. 2).

User interface models are often developed using user interface description languages (UIDL). Garcia (2009) defines a UIDL as "a high-level computer language for describing characteristics of interest of a UI with respect to the rest of an interactive application" (Garcia, 2010, p. 18). Puerta and Eisenstein (1999) identify five types of models which are necessary for UI development: task model, domain model, user model, presentation model, and dialog model. Of these, the dialog and presentation models are directly associated with the user interface itself. The presentation model specifies the visual, haptic, and auditory elements of the UI while the dialog model specifies how "the presentation model interacts with the user" (Puerta & Eisenstein, 1999, p. 172). Another common type of model used for representing UIs is the device models, which captures how UIs manifest across different types of mobile devices. Additionally, Sousa et al. (2010) define four levels of UI components:

> Screen Group: "a group of closely related screens and possible sub-groups to precisely classify screens"
> Screen: "a state of the user interaction where it is possible to perform a task or part of a task or even several tasks"

Screen Fragment: "a container of related elements in the screen"
Screen Element: "the most atomic component to perform user tasks (e.g., input or display, navigate)" (Sousa et al., 2010, p. 3)

As with BP and IS modeling, numerous UIDLs exist to support different purposes across different levels of abstraction (Engel et al., 2014). A common framework for categorizing these levels of abstraction is the Cameleon Reference Framework (CRF), with levels including "Model level, Abstract User Interface (AUI) level, Concrete User Interface (CUI) level, and Final User Interface (FUI) level" (Calvary et al., 2003; Engel et al., 2014, p. 184; Sousa et al., 2009). The "model" level of abstraction, also known as the "concepts and task model" level, can include a domain model (concepts) (e.g., UML class diagram) and task models necessary for providing the domain and use context for the lower levels of abstraction (Calvary et al., 2003). The AUI can be divided between "its static structure (the presentation model) and dynamic behavior (the dialogue model)" (Calvary et al., 2003, p. 18). Engel et al. (2014) provide a review of common UIDLs, including identification of the types of models and CRF abstraction levels supported by each (see Table 2).

Table 2. Review of Common UIDLs, Adapted from Engel et al. (2014).

Language	Types of Models	Abstraction Level(s)
User Interface Markup Language (UIML)	Presentation, Dialog, Domain	Model
User Interface Extensible Markup Language (UsiXML)	Task, Domain, Presentation, Context of Use (User, Platform, Environment), Mapping, Translation, UI	Model, AUI, CUI, FUI
Dialog Modeling Language (DiaMODL)	Domain, Dialog	Model
Interface Specification Meta-Language (ISML)	Task, Metaphor, Presentation, Domain, Dialog, Device	Model
Transformation Environment for Interactive Systems Representations (TERESA) XML	Task, Domain, Dialog, Presentation, Device	Model, AUI, CUI
Model-based Language for Interactive Applications (MARIA)	Task, Domain, Presentation, Event, Dialog	Model, AUI, CUI
Extensible Interface Markup Language (XIML)	Task, Domain, User, Presentation, Dialogue	Model, AUI, CUI
XML User Interface Language (XUL)	Presentation, Dialog	CUI

Modeling integrated business processes down to the UI level requires decomposition of business process models from a high-level perspective to the task models which are required for supporting user-interface design (Kolb et al., 2012; Kristiansen & Trætteberg, 2007; Pintus et al., 2010; Pontico et al., 2006). Task model elements are then associated with UI components specified in the UI model(s), as illustrated in Figure 23. An example of this progression, and the role of different modeling languages throughout that process, is presented by Sousa et al. (2009), as seen in Figure 24. Here a business process modeling language is coupled with the UsiXML, which supports both task modeling and UI modeling (see Table 2). While the AUI level of the Cameleon Reference Framework is not explicitly mentioned, it would ideally occur between specification of the task model and CUI model. "In the model-based approach, design progresses from task models, through dialog models to concrete interaction design, in a top-down process" (Trætteberg & Krogstie, 2008, p. 89).

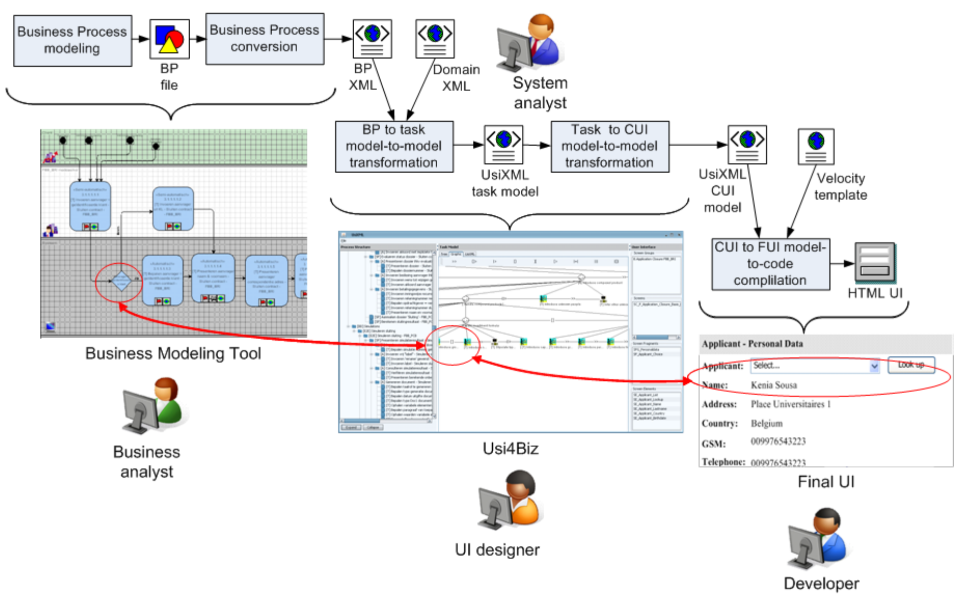


Figure 24. Method for Aligning a Business Process with Associated User Interface(s) and Task(s). Source: Sousa et al. (2009).

It is important to distinguish the more general development of business process, task, and user interface conceptual models from the model-based user interface development (MBUID) approach for automated user interface development. With the former, human readable conceptual models can guide the design of integrated business processes, information systems, and user interfaces with some level of abstraction (Butler et al., 2000; Trætteberg, 1999), supporting categories 1 and 5 of process model usage specified by Krogstie et al. (2008). Under the MBUID approach, tightly coupled computer-readable modeling languages support an automated translation of more abstract business process models into more precise (or concrete) user interface models and executable code (Limbourg et al., 2004; Puerta & Eisenstein, 1999). While MBUID supports automated development and adaptation of user interfaces to business process changes, the highly structured languages required for the tight coupling of MBUID models are generally inaccessible to all stakeholders (Pontico et al., 2006). In either case, ensuring the alignment of the process model(s), task model(s), and user interface (dialog) model(s) requires the creation of a domain model for the UI attributes and target domain (Kovacevic, 1999; Trætteberg & Krogstie, 2008). A common domain model should support semantic interoperability across both functional and user interface components of information systems and "provide a common language for specifying, visualizing, and documenting software artifacts" (Kovacevic, 1999, p. 258). "A domain model captures concepts from the semantics of the application domain. Without domain concepts a UI description would be an empty shell" (Limbourg, 2004, p. 38).

Domain models should capture concepts for all application domains necessary for ensuring common understanding across all stakeholders. This can include both the environment and processes or procedures an application will support (e.g., modeling the relationship between different system users and artifacts) and concepts within software domain itself which will need to be captured in the task or UI models. An examples of an ontologies which can support the latter component of domain models include the Core Software Ontology (CSO), which provides a common reference for software engineering concepts, and the RPA ontology which was recently developed by Völker and Weske (2021) to extend CSO for RPA considerations (Oberle et al., 2009; Völker & Weske, 2021). As discussed earlier, multiple ontologies can be aligned through the use of an upper ontology to avoid conflicts in semantics (Walter & Ebert, 2009). In addition to integrating separate domain ontologies for a consistent domain model, the user interface modeling also requires mapping multiple other models. Puerta and Eisenstein (1999) identify particularly important model mappings for UI design as including task-dialog, task-presentation, domain-presentation, task-user, task-domain, and presentation-dialog mappings.

The challenge of mapping these different models, or more generally the mapping of “abstract models” (e.g., task models) and “concrete models” (e.g., GUI dialog models), is known as the mapping problem (Puerta & Eisenstein, 1999). Different techniques exist to support the mapping of task and UI models, with most targeted for supporting UI design and development (Sousa et al., 2010; Vanderdonckt, 2005). The mapping problem can include mapping between different levels of abstraction (inter-level mapping) or within the same level of abstraction (Limbourg, 2004). For user interface designers mapping models in support of model-based user interface development,

> the expected win is that when a business process will change, the task model will change accordingly and so does the UI model corresponding to this task model, implementing a consistent alignment between business and user interfaces, and, consequently, providing traceability. (Sousa & Vanderdonckt, 2011, p. 126)

The automated development of software and user interfaces supported by MBUID is outside the scope of this research, which focuses on conceptual modeling approaches which support communication across varied stakeholders (category 1 of process model usage from Krogstie et al. (2008)) and guide the design of complex systems rather than automating their generation (category 5 of process model usage). Alignment of user interface conceptual models with business processes and user tasks is necessary for modeling the envisioned sim-IS environments, however, and lessons can be learned from related efforts in the MBUID domain. Traetteberg and Krogstie (2008) propose bridging the mapping problem by using BPMN for both process and task modeling, then using the DiaMODL UIDL for UI dialog model development, with a common domain class diagram supporting the alignment of the models. They describe the process of transitioning from process model (in BPMN) and class diagram (in Ecore domain modeling language) to user task model (also in BPMN) to user interface model (in DiaMODL).

Researchers in the MBUID domain have also attempted to address the mapping problem by extending business process modeling notations to support task and user interface modeling. Kovacevic (1999) propose an UML extension for enhanced task modeling in support of UI models. Auer et al. (2009) present a BPMN extension for user interface modeling, though the extension is specifically designed to support representation of submit/response-style UI interactions. Their approach leverages the “model decomposition and refinement mechanism in BPMN” to support multiple levels of abstraction (Auer et al., 2009, p. 370). The highest level supports business process modeling while the lowest level supports submit/response UI interaction modeling using BPMN extensions based on a different modeling approach (formchart diagrams).

Another example of a model-based approach to user interface development, one which is particularly relevant for this research, is found in the development of UIs for workflow management systems. Workflow management systems were developed in the 1990s to support automation of office work (van der Aalst et al., 2018). WFMSs are also considered a precursor to modern RPA software, a relationship which will be addressed later in this chapter. To guide the design and adaptation of WFMSs to changing integrated business processes, conceptual modeling methodologies were developed to map WFMS user interfaces to workflows, processes, and tasks. As with business process and task models, workflow and task models can often be used for the same domain, but at different levels of abstraction (Trætteberg, 1999). In a methodology for WFMS UI design presented by García (2010) (see Figure 25),

> a workflow is recursively decomposed into processes that are in turn decomposed into tasks. Each task gives rise to a task model, whose structure, ordering, and connection with the domain model allows a semi-automated generation of corresponding user interfaces by a model-to-model transformation. (Garcia, 2010, p. 5)

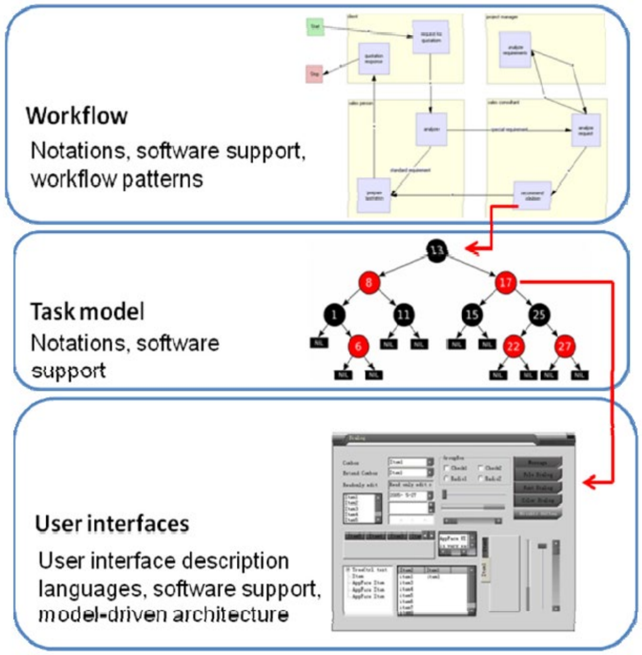


Figure 25. Decomposition of Workflows into Process, Task, and User Interface Models. Source: Garcia (2010).

Garcia (2010) presents a meta-model to guide the decomposition of workflow processes and tasks, and integration of their associated models, to inform the design of UIs. This approach, while created to support the development of UIs, would also be beneficial for the documentation of existing UIs as they relate to workflows and associated processes and tasks. What is notable in this approach is the way the UIDL, UsiXML, is extended to align task and UI models for a WFMS with business process and organizational models.

The structure of UsiXML aligns with the Cameleon Reference Framework (it actually derives from the same project) and it was developed to support the multiple levels of detail and abstraction necessary for the modeling of user interfaces, including task, domain, presentation, dialog, and context modeling (Engel et al., 2014; Limbourg et al., 2004). The result of the UsiXML extension by Garcia (2010) is a new modeling language called FlowiXML (user interfaces to workflow based on UsiXML), which extends UsiXML to support the alignment of process, task, UI, and organizational models. The key to this methodology is the mapping model which explicitly aligns the different models. Mapping models define the integration of models, whether the conceptual model consists of a single type of modeling language or heterogenous modeling notations. "Rather than proposing a collection of unrelated models and model elements, this proposal provides a designer with a set of pre-defined relationships allowing a mapping of elements from heterogeneous models and viewpoints" (Garcia et al., 2008, p. 8).

In situations where information systems aid users in their management of, or interactions with, complex systems, an additional type of model is important for the design of business processes, ISs, and UIs: mental models. Cognitive task analysis (and CWA) methods help designers develop an understanding of user mental models and their relationship with the IS and work environment for joint cognitive systems. As business processes, ISs, and UIs coevolve over time, it can be expected that the users' mental models will coevolve as well. This is the moving target problem discussed earlier and addressing it necessitates the development of explicit mental models for system users and continuous maintenance to ensure their alignment with business process, IS, and UI models (Sousa et al., 2009).

c. Mental Models for Sociotechnical Systems

Mental models have long been recognized within the IS community as critical for supporting the design of complex systems and the development of expertise in cognitively complex systems (Norman, 1983; Puerta-Melguizo et al., 2002; van der Veer & Puerta Melguizo, 2002). Norman (1983) identified four concepts which are useful for facilitating such discussion of mental models: the target system, system conceptual model, system user's mental model, and scientist's conceptualization of the user's mental model. Up until this point our discussion of business process, IS, and UI modeling relates to the system conceptual model concept. The user's mental model is an abstract representation of a system, which is continuously changing as a user interacts with a system, and is not directly accessible to researchers. Conceptualizations of user mental models are useful for supporting training design (i.e., guiding the development of trainees' mental models to achieve some target mental model) and system design (i.e., achieving an ecological interface design).

> Cognitive modeling has not yet played much of a role in the study of sociotechnical systems. Arguably, this is because most cognitive modeling systems were originally created to model microcognitive results, not the types of macrocognitive behaviors that drive sociotechnical systems (Klein et al., 2003). However, this does not mean that cognitive modeling systems cannot be adapted to deal with macrocognitive activities in ways that are relevant to cognitive engineering. (West & Nagy, 2007, p. 186)

Early work in cognitive modeling focused on microcognitive considerations, such as identification of declarative knowledge required for completion of routine tasks and anticipated workload associated with specific cognitive tasks. At this level of cognitive modeling, techniques such as HTA and GOMS are used to "map mental mechanisms at a microcognitive scale onto specific tasks that [center] on interaction with computers and computerized devices" (Hoffman & Militello, 2009, p. 76). As researchers began exploring more cognitively complex systems, most notably operations center actions relating to the Three Mile Island accident, the limitations of microcognitive approach became clear (Hoffman & Militello, 2009). In another oft-used example of macrocognition research, studies of novice and expert weather forecasters has found that weather forecasters' actions were based on the current weather situation and could not be modeled as a single "sequence of reasoning operations and strategies" (Hoffman & Militello, 2009, p. 204).

> Although we might want to reveal specific causal sequences of various memory or attentional mechanisms, this turns out to be difficult. When we try to describe naturalistic decision making, we quickly realize that it makes little sense to concoct hypothetical information processing flow diagrams believed to represent causal sequences of mental operations, because they end up looking like spaghetti graphs. (Klein et al., 2003, p. 81)

Despite some efforts to build models of macrocognitive phenomena from a microcognitive perspective, this is often an unreliable and context-dependent approach (Hoffman & Militello, 2009; West & Nagy, 2007). Instead, modeling macrocognitive functions in STSs often requires approaching such functions from the macrocognitive perspective from the beginning, rather than attempting to scale up microcognitive models associated with individual task models. This includes use of CTA methods for identification of how personnel with different levels of expertise conceptualize the underlying structure and dynamics of a system or domain and what cues they use for informing their analysis of those systems. Despite their general disconnection from microcognitive functions, CTA methods provide greater rigor for cognitively complex tasks at the macrocognitive level.

While user mental models can be developed and evaluated in consideration of both microcognition and macrocognition, they include different levels of abstraction in the different contexts. For example, for well-defined, routine tasks, user mental models can be compared to conceptual models developed by system designers, to evaluate user understanding of tasks and the appropriate use of UI elements for task completion. This would constitute a microcognitive approach, as the researcher would essentially be evaluating the declarative memory of the user for well-defined processes. In a macrocognitive context, user mental models are ideally acquired in support of training or system design by extracting expert mental models through CTA methods. In the absence of true experts, such as within the military or when a novel system is introduced, the designers' conceptual model of the envisioned sociotechnical system may provide a starting point.

As discussed at the beginning of this chapter, another tool for supporting IS design in cognitively complex domains is CWA, which can support design of ecological interfaces. While CTA methods provide an understanding of cognitive tasks that are expected or known to be required, CWA methods provide an understanding of the work environment. CWA methods provide a more abstract representation of a work environment and include "methodologies that can identify the fundamental cognitive and collaborative demands of the work domain that transcend particular technologies and interfaces" (Roth et al., 2001, p. 134).

> Microcognitive models present causal-chain understandings of mental events, built from mental operations such as short-term memory access and attentional shifts…On the other hand, macrocognitive models describe the major goal-directed functions of cognitive work (deciding, replanning, sensemaking, problem detection, and so on) and the cognitive processes that support those functions (for example, developing mental models and maintaining common ground)…The creation of macrocognitive models of aspects of sociotechnical work systems could help systems and software engineers as well as cognitive systems engineers develop high-level understandings of the nature of the cognitive work. (Hoffman, 2012, p. 155)

Each of these levels of cognitive modeling are important for different purposes. For task models, and their alignment with UIs, cognitive modeling from the microcognitive perspective can identify the declarative knowledge which is necessary for each task and where users are expected to access each piece of information. Such a model can then be used to guide the design of low-level user training or to design an RPA bot to emulate user-system interactions. This can include identification of what information the RPA workflow must acquire, and from where, and what cognitive processes must be emulated in the manipulation of said information prior to entry in another IS interface. When determining how to design a sim-IS environment to support the development of expertise in a domain, mental models developed through CTA methods can aid sim-IS environment designers in determining the requisite system dynamics to simulate and how to present them to trainees through context-appropriate cues that develop target mental models. Different sociotechnical systems which may require consideration by decision-makers are discussed in the next section. Personnel with different levels of expertise often rely on different types of cues from the operating environment and different ways of explicitly representing the relationships between those pieces of information (Hoffman et al., 2014). While both microcognitive and macrocognitive-level cognitive models may be necessary for modeling an integrated business process, unlike other modeling approaches discussed here, decomposition of mental models from higher to lower levels of abstraction is rarely feasible.

Cognitive models at different levels of abstraction between macrocognition and microcognition can also be associated with the broader sociotechnical system models at different places across the business process, IS, task, and UI models. Cognitive models at the microcognitive level can be tightly aligned with task and UI models, identifying the knowledge and cognitive actions required for specific tasks in association with specific UI elements. Cognitive models at the macrocognitive level are particularly important for understanding users' understanding of complex cognitive systems at the higher levels of abstraction. One methodology for macrocognitive modeling, the macrocognitive modeling procedure (MMP), is intended to "help the systems engineer bridge the gap between his or her needs and those of the cognitive systems engineer" by providing a "base model of expert reasoning" which can be adapted to different domains (Hoffman, 2012, p. 156).

Association of user mental models with conceptual models of integrated business processes are also necessary for anticipating changes in user mental models as integrated business processes evolve over time. Mental models can capture a user's understanding of how to manipulate the IS to accomplish a task (linking task and UI models) and how the information presented by the IS relates to relative structure and dynamics of the broader STS (linking UI, IS, and business process models). As business processes, ISs, and UIs coevolve, the user's understanding of their tasks and the context of those tasks in the integrated business process must also evolve.

> If we can predict or understand even in some fashion what mental models a new operator or user might hold about a system and its relevant domain, and what model they might build through subsequent interaction with the system, then we can improve interface design, training, operating procedures, and so on. By understanding the potential users' mental models, and by adapting their own conceptual model accordingly, designers might develop a system image that better matches, sustains and helps develop an appropriate user mental model. (Charles et al., 2015, p. 394)

The continuous coevolution of integrated business processes necessitates an adaptive training program and an adaptive sim-IS integration methodology to support trainees in the face of such continuously evolving operational environments (Kang & Santhanam, 2003). This presents a challenge for training organizations who must be prepared to refine the sim-IS environments as the referent (i.e., the business processes, ISs, and BP-IS dynamics) continuously evolves. The challenge of achieving BITA among "permanently changing process structures" (Millet et al., 2009, p. 400), and the alignment of personnel training with continuously evolving integrated business processes, is similar to the dynamics of JCS coevolution addressed by the CSE community. Growing interest in the application of artificial intelligence in human-machine teams presents an additional way documentation of user mental models in context present an important consideration for the design of systems and training environments. Hoffman et al. (2019) distinguish between "mental models," which refer to a user's understanding of a domain or system, and “user models,” which refer to a computer’s understanding of the user’s mental model (Hoffman et al., 2019, p. 9).

Despite recognition of the importance of BP-IS conceptual modeling, many organizations lack defined, standardized processes for leveraging their information systems. This can present an obstacle for developing target user mental models to inform training design. The design and management of such BP-IS conceptual models can be particularly challenging for military and emergency management organizations, where day to day ‘garrison operations’ are different from operations in a deployed environment. This can result in a sort of ‘chicken or egg’ problem. Defining and standardizing integrated business processes requires an environment for exercising and evaluating the resulting BP-IS model, but such an environment is not available until units deploy. ***One way to overcome this problem, and begin the cycle of continuous BP-IS coevolution, is to standardize a BP-IS model using what knowledge is available, and begin exercising and improving it in sim-IS environments.***

d. Sociotechnical System Dynamics

The Dynamic Model of Situated Cognition, introduced earlier, highlights the importance of understanding perturbations of information as it propagates through a STS from the sources, be they humans or technological sources, to the recipients who use the information to inform decision-making. Understanding the ways information can be degraded is important for both the design of integrated business processes and the design of training for development of appropriate user mental models. This section identifies different types of information degradation to be represented in sim-IS environment conceptual models in support of system design and user training.

Multiple IS models support description of the value of ISs in context (DeLone & McLean, 2003; Gable et al., 2008). The DeLone and McLean model of IS success specifies six dimensions that influence IS success: information quality, system quality, service quality, intention to use, user satisfaction, and net benefits. The IS-Impact Measurement Model identifies four general categories (individual-impact, organizational-impact, system-quality, and information-quality), with several of the measures associated with system and information quality mirrored in the updated DeLone and McLean Model of IS Success. Several of these information and system quality measures can be influenced by the nature of IS integration in an operating context, such as data accuracy (associated with both system quality and information quality), data currency (system quality), or timeliness, completeness, and precision (information quality).

There are many ways information may be degraded as it propagates through a sociotechnical system. One type of information degradation which is outside the scope of this research is the degradation associated with the loss of data exchanged across communication systems and networks. Commonly referred to as "communications effects" (Hieb & Timian, 1999), this includes information degradation resulting from issues like "radio propagation loss and bandwidth utilization" (Bailey et al., 2004, p. 867). In a 2004 article expounding the need for representing these effects through communications effects servers, uniformed U.S. Army personnel and consultants found that "most current Army testing, training, and experimentation events are planned and executed assuming perfect communications, with no restrictions, such as latency or bandwidth, that would be encountered in the real world" (Bailey et al., 2004, p. 867).

Bailey et al. (2004) go on to describe communications effects as consisting of "propagation effects" and "network effects." Propagation effects are defined as "the factors that affect the transmission of electromagnetic signals, such as terrain, foliage, buildings, and the atmosphere" (Bailey et al., 2004, p. 868). Network effects are defined as "the effect of network organization, routing, and network performance on data transmission" (Bailey et al., 2004, p. 868). Communications effects servers, such as the Aggregate Level Communications Effects Server (ALCES), Joint Network Emulator (JNE), and Network Effects Emulation System (N2ES) simulate these kinds of communications effects within individual simulations or in sim-C2 information exchange (Pullen & Ruth, 2018a; Wilson, 2017). Communications effects servers can use standard protocols like HLA and DIS for simulating communications effects within an individual simulation, across multiple simulations, or in the exchange of information between a simulation and C2 system (Bailey et al., 2004). However, communications effects servers are not equipped for simulating the entry of information in HitL ISs. They are limited in their ability to support representation of different types of information degradation, particularly those related to HitL data entry in ISs and data exchanges which lack the standard protocols used by simulations and C2 systems. This limitation is due in part to the difference between communications systems and information systems.

> Communications systems are used to interconnect information systems – the computers and software that warfighters rely on for information…communications models must be integrated with information systems models to form a single comprehensive representation of network centric warfare. (Bailey et al., 2004, p. 872)

A similar type of information degradation across communication systems (and information systems) can be found in cyber-attacks and electronic warfare, where information degradation is imposed intentionally by enemy forces. Cyber-attacks and electronic warfare can range from the modification and deletion of information to more overt denial of service, jamming, spoofing, and ransomware attacks. Pullen and Ruth (2018) propose modifying C2SIM servers, with the addition of what they call a "cyber effects editor" to simulate cyber-attacks "by modifying or deleting messages passing through C2SIM servers" (p. 5). This approach, with its dependence on the C2SIM protocol-based sim-C2 information exchange, is limited in its ability to simulate cyber-attacks on many ISs, like logistics ISs, which do not use the common C2 protocols for information exchange. Communications effects servers and envisioned cyber effects editors support simulation of information degradation due to propagation effects and cyber-attacks across communications systems. While they do not address how the nature of technology integration in business processes affects the propagation of information across STSs, and are therefore outside the scope of this research, simulation of these dynamics for HitL IS such as logistics IS is deserving of future research for RPA-based sim-IS environments. This research is focused on those STS dynamics resulting from the performance of ISs, similar to what Bailey et al. (2004) call "information systems modeling" (p. 872), and the nature of IS integration in organizations' integrated business processes. It is notable that of the three general types of information degradation discussed here (communications effects, cyber-attacks/electronic warfare, and information system degradation), the two which are currently supported with means of representation in sim-C2 environments are those which are due largely to external factors.

It is beyond the scope of this research to identify of the all dynamics associated with information propagation across STSs, or how they can all be simulated in a sim-IS environment. Instead, we address a few types of information and system qualities, roughly categorized as consisting of temporal information exchange qualities (timeliness, frequency, and latency of information exchange) and information content qualities (data accuracy, precision, and completeness). Our intent is not to provide a holistic taxonomy for STS information propagation dynamics, but to identify a few key dynamics whose representation in a sim-IS environment may benefit design of systems and training environments. Definitions for the temporal dynamics of information propagation through STSs are informed by numerous efforts by researchers and practitioners to understand the time-value of information and evaluate the benefits of investments in real-time IT (Cundius & Alt, 2013; Hackathorn, 2004; Polites, 2006).

It is common for commercial and military organizations to assert the importance of having timely and accurate information to support decision-making (Davenport & Snabe, 2011; Lurie & Swaminathan, 2009; Stuetelberg & Thomas, 2021). Increasingly capable computerized information systems are often acquired or developed with the expectation that they will assist in improving one or both of these attributes for an associated integrated business process. An example of this is found in the advent of in-transit visibility (ITV) technology, including radio-frequency identification (RFID), in commercial and DOD supply chain management. Walmart was an early leader in the implementation of RFID technology to provide timely and accurate insights regarding the locations of pallets and cases of products, increasing the efficiency and responsiveness of their supply chain management (Reyes et al., 2016). The DOD has also invested in RFID technology to enhance asset visibility. With the technological advancements, however, it has become evident that real-time IT alone is not enough, and the value of real-time IT can only be achieved if paired with suitable business processes (Cundius & Alt, 2013, 2017; Davenport & Snabe, 2011). Cundius and Alt (2013) propose a real-time assessment model to assist in determining the “real-time level” of integrated business processes for which real-time IT is considered. Several of the same factors are critical for informing decisions regarding investment in real-time over near real-time ISs, supporting their real-time assessment model, they can also be viewed as important for describing key temporal dynamics of integrated business processes in support of both system and training design.

The Action Distance Model, seen in Figure 26, defines latency in terms of data, analysis, and decision latency and the impact of each on the value of information as it propagates through an STS. Hackathorn (2004) refers to latency as "the time required to capture (usually from some source transactional system), transform/cleanse and store data" in an information system (or data warehouse) (Hackathorn, 2004, p. 2). Polites (2006) extends this definition of data latency to also include "the time required to scan the environment and search for information that comes from sources other than a transactional data warehouse" (p. 1388). This is a useful definition for data latency for the military logistics context, as it can be used to address the time required to manually count pallets of ammunition or consolidate reports of fuel statuses from subordinates before entry of the information into the appropriate logistics IS.

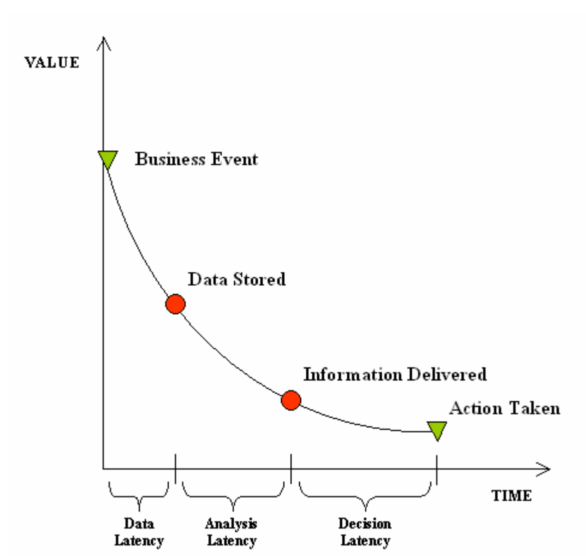


Figure 26. The Action Distance Model. Adapted by Polites (2006) from Hackathorn (2004).

Frequency refers to the number of times information is transmitted/updated within a certain period of time (e.g., updating statuses once a day). Timeliness is used to refer to numerous phenomena in the literature, and at time used interchangeably with latency (Cundius & Alt, 2013). We define timeliness as the time between information transmission and a scheduled deadline (e.g., reporting the number of MREs on a base two hours after the daily 1700 deadline) or a status change (e.g., when a commander's critical information requirement [CCIR] is tripped). In this research latency is used to refer to data latency, or the time between the acquisition/generation of information and its transmission/entry in an IS (e.g., reporting the number of MREs on hand at 1700 based on counts taken at 0500).

An additional temporal consideration for describing the propagation of information across STSs is the difference between what we refer to as continuous information exchange and event-driven information exchange. Continuous information exchange refers to information exchanges which are conducted routinely in accordance with some specified level of frequency or schedule. This can include hourly, daily, or weekly status reports as well as much more frequent, automated position updates provided by GPS-supported (GPS) technology, like BFT. Event-driven information exchange refers to information exchanges which are triggered by events or actions meeting some prespecified threshold (e.g., 80 percent of a critical asset destroyed). Event-driven information exchanges can include units' CCIRs. While this research focuses on information exchange dynamics associated with continuous information exchange, event-driven information exchange provides an important area for future research, particularly as CCIRs are intended to drive a commander's decision-making in response to key developments on the battlefield.

For the DOD, the frequency of information exchanges across distributed networks must also be weighed against a cost which is not considered by commercial organizations: increased risk of detection by enemy forces due to an increased electromagnetic signature (Stuetelberg & Thomas, 2021). On the modern battlefield, the transmission of data from ISs increases a unit's electromagnetic signature, which may be used by enemy forces to locate the unit's position. Military organizations must therefore identify how to decrease their transmission of logistics information without degrading their ability to respond to logistics requirements beyond an acceptable level. This issue an ongoing problem for the

Marine Corps logistics IS community. While this is outside the scope of this research, a sim-IS environment which simulates temporal dynamics of information propagation across STSs may benefit this analysis.

In this research, in addition to temporal information exchange qualities, information degradation is addressed for information content qualities in terms of accuracy (e.g., incorrect values entered in an IS accidentally or intentionally), precision (e.g., rounding values reported to varying degrees), and completeness (e.g., missing requisite data in reports). Additional considerations regarding the quality of information exchanged are the different perspectives and levels of specificity which can be expected at different times in a process. For example, equipment operators and maintenance personnel generally have different levels of expertise in diagnosing and reporting causes of equipment degradation. When a piece of equipment is damaged, either due to enemy action or non-combat damage, the equipment readiness reports which are initially generated, by equipment operators, may be less accurate (incorrect diagnosis of maintenance defect) and detailed (incomplete identification of repair part requirements) than the reports generated later when the equipment is inspected by maintenance personnel. Just as these different reports are generated by different personnel, they may be propagated through the organization through different information systems. Within the Marine Corps, while motor transport operators may update initial changes in equipment readiness information in the TCPT IS, for example, maintenance personnel will enter more accurate and detailed information in the maintenance IS (Morse, 2017). While this type of information degradation, in the form of different information perspectives, is not addressed directly in this research, it is an important consideration to be addressed during the conceptual modeling of information exchange for sim-IS environments.

A taxonomy of logistics functionality was proposed by Rybacki and Blackman (1997) for use in representing logistics considerations in military modeling and simulation. Their proposed taxonomy would support communication across stakeholders for both operational logistics ISs and simulations designed to simulate logistics functions. The taxonomy would consist of four components: processes, objects, algorithms, and data (Rybacki & Blackman, 1997). Given the importance of communications effects and other

STS dynamics on the effectiveness and efficiency of logistics integrated business processes, a fifth component should be considered for inclusion in such a logistics taxonomy to facilitate communication in the representation of these dynamics.

An additional challenge for DOD (and emergency management) organizations is the infrequent availability of real-world systems to measure. For emergency management and DOD organizations, where real-world implementation of integrated business processes occurs infrequently and in austere environments, the dynamics of existing integrated business processes can be difficult to measure. Field exercises offer some opportunities for measurement of these dynamics, but even these environments are significantly limited in their ability to represent real-world dynamics. Even month-long Marine Corps field exercises conducted aboard Marine Corps Air Ground Combat Center (MCAGCC), the Marine Corps' largest base, for example, are too limited in time and space to stress logistics processes (Morse, 2016).

D. ROBOTIC PROCESS AUTOMATION

Unlike most C2 systems which receive machine-to-machine messages using common protocols (i.e., inside-out interoperability), the processes for updating HitL ISs are conducted manually by human operators navigating graphical user interfaces. The automation of these processes, and the simulation of associated BP-IS dynamics, can be achieved through the "outside-in" interoperability approach supported by RPA technology. RPA presents an opportunity for a new, fourth means of sim-IS information exchange, one which offers modularity, reuse, cost efficiency, and higher fidelity for simulating sociotechnical processes. Before discussing how RPA can revolutionize sim-IS integration, however, it's worth discussing how it is disrupting conventional business processes. Unlike the "classical 'inside-out' approach" to integrating disparate information systems by modifying the systems themselves, RPA achieves system interoperability (and process automation) by mimicking a human user's interactions with the systems' graphical user interfaces (van der Aalst et al., 2018, p. 269). This "outside-in" approach enables organizations to automate processes across numerous, unconnected systems without making expensive engineering changes to the information systems themselves.

In their 2019 report on the RPA software market, Gartner defines RPA as "a digital enablement technology that predominantly leverages a combination of user interface and surface-level features to create scripts that automate routine, predictable data transcription work" (Miers et al., 2019, p. 1). The report identified RPA as "the fastest-growing software subsegment officially tracked by Gartner" in 2018, with market leaders being Automation Anywhere, Blue Prism, and UiPath (Miers et al., 2019, p. 41).

In order to implement RPA software for process automation, organizations first identify processes that have a sufficient degree of consistency, such that the process logic can be programmed, and regularity, making it cost efficient to build the RPA 'bot' for the task. Due to their use of existing user interfaces for interaction with an organization's information systems, the "outside-in approach," RPA bots are easier and more cost efficient to develop. This makes them more accessible for tasks which previously would not have been automated through an "inside-out" approach to system integration (see Figure 27).

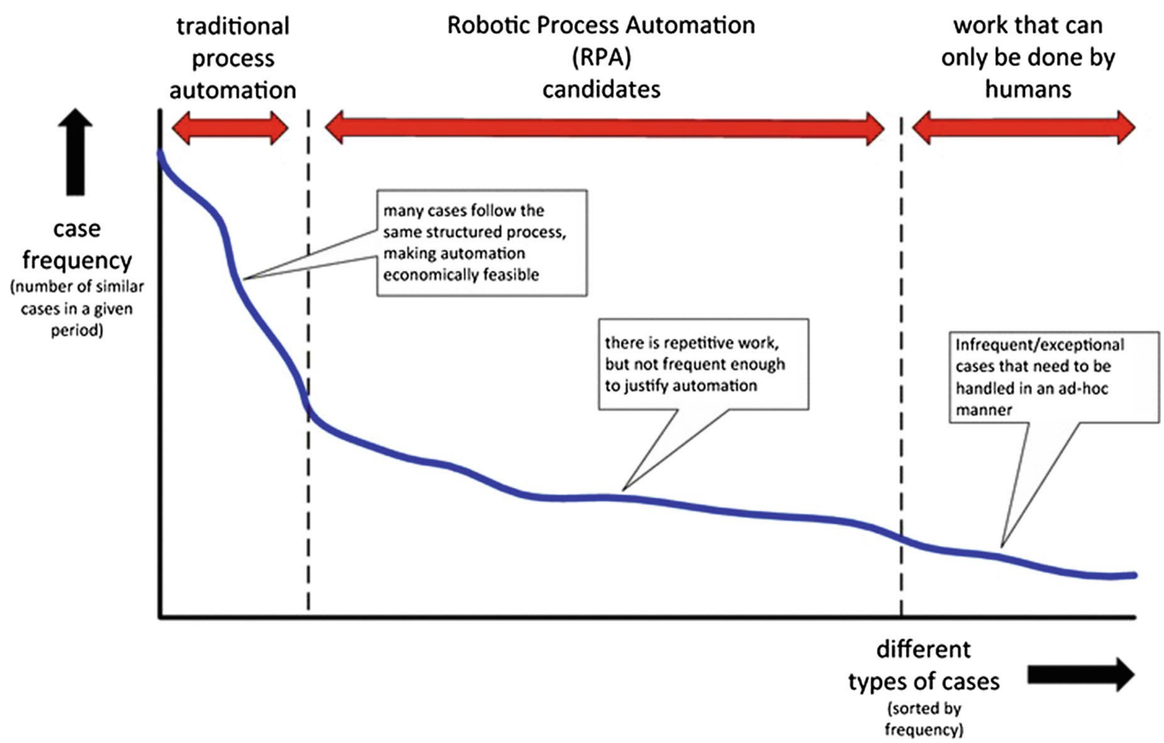


Figure 27. RPA Application Relative to Process Structure and Frequency.
Source: van der Aalst et al. (2018).

RPA software is implemented in either an attended or unattended mode. In the attended mode, a human monitors the RPA software's actions, and the RPA bot uses the human's credentials to access various information systems. In unattended mode, RPA bots operate without real-time human oversight and can be assigned their own credentials for accessing certain information systems. Another significant distinction between attended and unattended RPA bots is where they are generally hosted. Attended RPA bots are executed on user's desktops, and have often been used "in call centers, working side-by-side with a customer support representative" (Tornbohm & Dunie, 2017, p. 4). Most RPA services are run in unattended mode, where RPA bots are hosted on a server and deployed as required to support numerous clients across a company. In this unattended mode, a single RPA bot can iterate through requests, using a library of process scripts as depicted in Figure 28, or tasks can be divided amongst multiple RPA bots for concurrent execution.

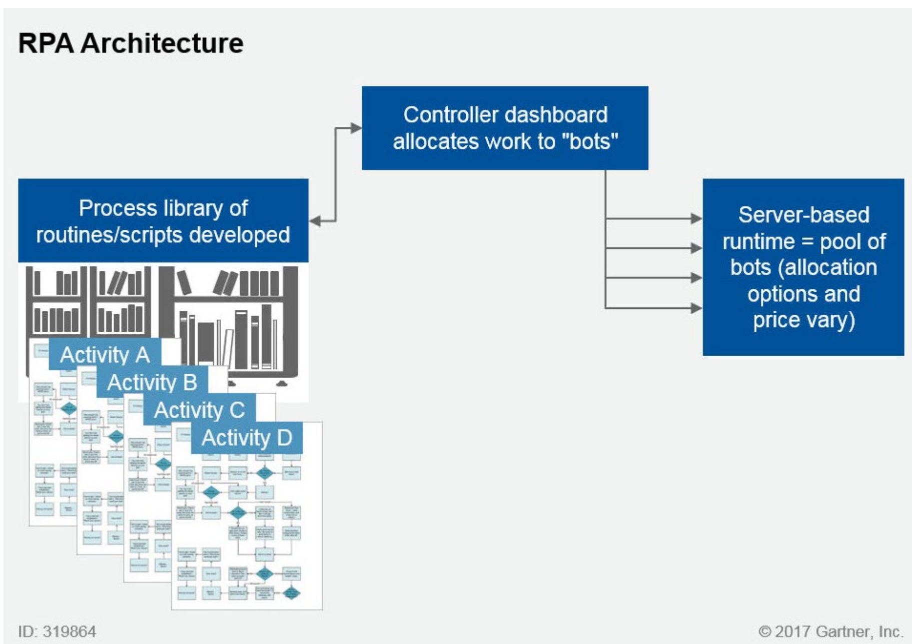


Figure 28. General Unattended RPA Bot Management Approach. Source: Tornbohm and Dunie (2017).

While RPA bots can be run directly from software on a user's desktops (e.g., UiPath's Community Edition), centralized RPA servers enable organizations to better manage their RPA tools. Under an RPA server-client approach, a centralized server dispatches RPA bots to client machines (or conducts requested actions locally) according to a consolidated schedule. This centralized scheduling and management of RPA capabilities ensures efficient RPA bot utilization and effective management of RPA workflow modeling and execution.

1. RPA Use in Business and U.S. Government

The automation of business processes is generally achieved through information systems known as business process management systems (BPMSs). This process automation often depends on the integration of a BPMS with disparate information systems across an organization, posing a technical challenge and limiting the implementation of BPMS technology (Dumas et al., 2013). The screen scraping and virtual keyboard/mouse techniques employed in RPA circumvent these limitations and allow companies to automate a broader range of processes.

RPA is being implemented across the federal government, and has been identified at the highest levels as a valuable supporting technology for efforts to "modernize operations" and "significantly reduce the burden on Federal employees" (Mulvaney, 2018, p. 5). In 2017, the National Aeronautics and Space Administration (NASA) became the first Federal agency to implement RPA (Constans, 2020). The NASA Shared Services Center (NSSC) employs RPA as one element of the Intelligent Automation Services (IAS), automating "low-value," repetitive tasks in order to enable employees to focus efforts on "more cognitively challenging and creative work" (LeMere, 2019, p. 8). By January, 2020, the recently established Federal RPA Community of Practice (CoP) consisted of over 50 Federal agencies and published the first version of its RPA Program Playbook (Federal RPA Community of Practice, 2020).

The DOD has made strides in implementing RPA for the automation of "low-value" work processes and as a part of artificial intelligence efforts. Leaders within the DOD for RPA implementation are the Office of the Under Secretary of Defense – Comptroller (OUSD-C), Financial Data Transformation Office, and the Defense Logistics Agency (DLA), both of which focused initial efforts on implementation of RPA to reform and enhance financial management processes (J. Felsted Sr., personal communication, September 20, 2019; E. Thomas, personal communication, September 13, 2019). At the beginning of FY2020, DLA also began implementing RPA, specifically UiPath, in an unattended mode.

RPA is also being implemented across the uniformed services. The Marine Corps Technology Services Organization (TSO) has established itself as a leader for RPA implementation within the Marine Corps (M. Sams, personal communication, October 23, 2019). While the TSO is focused on the application of RPA in support of financial management and accounting processes, like most organizations, they provide limited support to other Marine Corps units seeking to leverage RPA for other purposes. The Joint Artificial Intelligence Center (JAIC) also worked to develop RPA workflows with UiPath software, until it was dissolved.

2. Designing and Managing RPA Workflows

Primary obstacles to the implementation of RPA workflows include identification of processes suitable for automation (Leopold et al., 2018), designing platform-independent solutions (Völker & Weske, 2021), and maintenance of RPA workflows as processes and information systems evolve over time (Noppen et al., 2020). The first issue has been the subject of research proposing both automated and manual solutions to identifying tasks suitable for RPA. The second has, until recently, been impeded by the lack of a common, platform-independent domain model for describing RPA workflows. Research in the third issue has focused on the question of whether RPA workflow management should be centralized or decentralized and the extent to which it should reside in IT and other organizational units.

The low-code/no-code nature of RPA makes it highly accessible to non-technical personnel outside the IT department. This has resulted in varied approaches for RPA implementation and management within organizations. Models of RPA management range from centralized to decentralized to federated, with additional debate regarding whether RPA management is best positioned within IT department or elsewhere in organizations (Noppen et al., 2020). All of these models include challenges in maintaining RPA workflows over time. While the "loose coupling of RPA and IT [departments]" affords opportunities in the form of local ownership and responsiveness to business unit needs, it also presents additional challenges "related to lack of control and lack of end-to-end process view" (Osmundsen et al., 2019, p. 6926).

The challenge of maintaining end-to-end process view is compounded by a lack of platform-independent terminology and standards in the RPA domain (Völker & Weske, 2021) and the differences in modeling expertise between IT and business domain specialists. Bowen et al. (2021) identify the importance of identifying who creates task models: system users or system designers/developers. The different modelers have different areas of expertise and different levels of expertise in applying different modeling approaches. While business personnel outside the IT department may be familiar with some business process modeling notations, they are generally less familiar with the sort of software modeling approaches necessary for the management of software and its alignment with integrated business processes. Bowen et al. (2021) highlight a dearth of literature regarding task and UI modeling approaches from the user perspective, with researchers instead focusing on the role of modeling in support of automated UI development. While RPA workflow development does not necessarily require programming abilities, the design and maintenance of RPA workflows does require conceptual modeling of workflows in integrated business process context just as much as other software.

As discussed in the conceptual modeling section, changes in UIs and associated business processes can require adjustments in RPA workflows. This has been shown to result in long-term costs for RPA maintenance (Axmann et al., 2021; Noppen et al., 2020). Developing RPA conceptual models requires an understanding of the relationship between business process, task, and UI models as well as an RPA domain ontology to support the development of vendor-independent RPA conceptual models. Völker and Weske (2021) identify the need for a general RPA ontology to support vendor-independent RPA workflow modeling. They fill the gap by extending the core software ontology to support for RPA considerations. With this RPA ontology, and a domain ontology for the appropriate operating domain, RPA designers and managers can develop and maintain platform-independent RPA conceptual models.

Workflow management system conceptual modeling discussed earlier provides an example for approaching RPA workflow conceptual modeling. These conceptual models progress from high-level process/workflow models to lower-level tasks models, supported by a common domain model. Designing and maintaining such conceptual models is necessary for supporting RPA flexibility in adapting to changing business processes and information systems, as seen with workflow managements (Stavness & Schneider, 2004). What differentiates RPA modeling from WFMSs modeling is the need for task models to be associated with UI models to capture specific interactions with UI interface components, as UI models capture the primary means of information exchange between the RPA workflow and different ISs. RPA conceptual models can facilitate the design of vendor-independent RPA workflows, their adaptation relative to business process and UI changes, and the transition between RPA vendors over time as required by the organization.

E. LITERATURE REVIEW SUMMARY

This chapter presented the state of research and practice across several domains directly related to this research. This included a discussion of the design and development of simulation-supported, scenario-based training environments for integrated business processes, the design and development of integrated business processes themselves, and the state of robotic process automation technology. The first section identified the importance of simulation-supported, scenario-based training environments in supporting the development of ERP competencies for NDM environments. The capabilities and limitations of existing approaches for sim-IS environments are presented, with an emphasis on situated cognition and NDM considerations for the design of simulation-supported training environments.

The JCS perspective was presented next as an important consideration for framing the challenge of designing training environments which adapt to ever changing integrated business processes and associated user competencies. This perspective highlights the importance of designing an RPA-based sim-IS architecture which is modular and capable of evolving as the target integrated business process and STS dynamics evolve. It also highlights the importance of presenting a DSEEP overlay which supports such an iterative coevolution of sim-IS environments with the simulated JCSs, as discussed in Chapter VII. This section also briefly identified the potential future value of sim-IS environments which support the holistic evaluation and development of JCSs, an issue worthy of exploration as future work.

The discussion of distributed simulation design and development standards and broader simulation and IS conceptual modeling considerations in the third section provides context for the DSEEP overlay recommendations presented in Chapter VII. The exploration of conceptual modeling challenges across the IS, GUI, and STS domains illustrates the many important issues to be considered in the design of sim-IS environments for simulation of integrated business processes. It also highlights the risks associated with attempting to develop a prescriptive standard for machine readable conceptual models for simulation of STSs. This section also identified the influence of various STS dynamics, such as latency and timeliness of information, on the employment of ISs in context and the underrepresentation of such STS dynamics in sim-IS information exchange mechanisms which are more focused on the representation of network and propagation effects.

The state of RPA is presented here in terms of both the implementation and the design and management of the technology. While RPA is commonly referred to as a tool for "simulating" the actions of human operators of ISs, RPA practitioners are rarely interested in simulating the nature of human-IS interactions. This research explores how RPA can be used to support the simulation of human user interactions with ISs in the broader context of simulating STS dynamics. The challenge of modeling RPA workflows with platform-independent conceptual models, addressed in the second half of the RPA section, is directly related to this issue of simulating human-IS interactions and the challenge of modeling of those interactions, as addressed earlier in this chapter.

This issue of modeling human-IS interactions in context is a common thread across the different domains explored in this chapter. Developing user competencies requires an understanding of the integrated business process and the requisite user competencies and mental models to be developed. For sim-IS environments to support such training, sim-IS environment conceptual models must be informed by models of the target integrated business processes. This in turn informs the design of sim-IS information exchange mechanisms, such as RPA-based sim-IS mechanisms explored here. The JCS perspective highlights the need to continuously refine models of the target integrated business process(es) and the sim-IS environments through which they are simulated and exercised. The different fields of research explored in this chapter present important perspectives for the design of the RPA-based sim-IS environments and the process through which such environments are designed and developed.

III. RESEARCH DESIGN

This research consisted of three major phases: development of an RPA-based architecture and prototype workflow modules for automated sim-IS information exchange and simulation of BP-IS dynamics, evaluation of the RPA-based architecture and prototypes, and development of guidelines for the design and development of RPA-based sim-IS environments. The first two phases included iterative development and evaluation of entity and aggregate constructive simulations, and provided insights for the third phase. This research addresses the limitations of existing sim-IS information exchange approaches on two primary levels: *automating* sim-IS information exchange through RPA and *enhancing* sim-IS information exchange through RPA. The RPA-based sim-IS architecture was designed to provide a lower cost, modular means of automated sim-IS information exchange than conventional information exchange approaches. The enhancing of sim-IS information exchange includes the simulation of information delay and degradation found in sociotechnical systems. An RPA-based sim-IS information exchange mechanism prototype was be developed and evaluated in support of these research objectives.

The first phase of this research began with an exploration of RPA technology as a means for achieving an "outside-in" approach to sim-IS interoperability, with RPA "bots" effectively emulating puckster actions as they interact with simulation and IS graphical user interfaces to effect the transfer of information between simulations and HitL ISs. By decreasing the cost (in time and manpower) for including such HitL ISs in staff training environments, the proposed RPA-based method can facilitate an increase in the number and variety of ISs represented in sim-IS environments. This research included the design of a modular architecture for RPA workflow modules which supports the reuse of RPA modules associated with disparate simulations and ISs, and which supports modification as the associated JCSs evolve over time. This phase continued with the development of RPA modules to simulate the dynamics of integrated business processes unsupported by conventional sim-IS information exchange approaches, i.e., simulating the current or anticipated delay and degradation of information propagating through sociotechnical systems to the information systems.

Throughout the development and evaluation of the RPA-based information exchange architecture and prototypes, lessons were learned which informed the third phase of this research, filling a gap in existing standards for the design and development of distributed simulation environments. DSEEP does not currently provide guidance regarding the design and development of sim-IS information exchange mechanisms to represent the information exchange dynamics of referent sociotechnical systems. This research provides recommendations for the development of a sim-IS DSEEP overlay, including considerations for the development of sim-IS conceptual models which address the use of BP-IS and CTA models as referents for the design of information exchange across sim-IS environments. These guidelines explore how sim-IS designers/developers can leverage IS conceptual models, business process models, CTA, and simulation conceptual models to develop sim-IS conceptual models and data exchange models. They also address how such sim-IS conceptual models guide the development of RPA workflows (or other information exchange mechanisms) to simulate target sociotechnical system dynamics.

A. RESEARCH QUESTIONS

1. **Primary:** How can an RPA-based "outside-in" approach to sim-IS information exchange simulate temporal and content quality degradation of information across sociotechnical systems?
2. Subsidiary: How can an RPA-based "outside-in" approach to sim-IS information exchange facilitate automated exchange of information between simulations and HitL ISs in support of training exercises?
3. Subsidiary: How can RPA-based middleware and data exchange models for sim-IS information exchange be designed to facilitate modularity and reuse across different sim-IS environments?
4. Subsidiary: How can sim-IS environments be designed to address the dynamics of BP-IS integration in sociotechnical systems?

B. RESEARCH SCOPE AND ASSUMPTIONS

This research addresses the design and development of RPA-based middleware for automation of sim-IS information exchange and simulation of BP-IS dynamics. To accomplish this, generic conceptual models of integrated business processes (e.g., reporting fuel statuses via a specified IS) were developed in coordination with the MCLOG staff, the primary stakeholders for this research. These models defined the information exchange processes to be simulated and the nature of the information delay and degradation represented in those processes.

1. Scoping Robotic Process Automation

RPA technology is often defined as a broad class of software which automates tasks by operating on the same user interfaces that human users do (Fernandez & Aman, 2018; Hindel et al., 2020; van der Aalst et al., 2018). RPA technology can also be developed as an organic function for the software upon which it operates or coupled with AI/machine learning techniques (van der Aalst et al., 2018). This research considers RPA software defined as stand-alone software which supports the automation of rigid procedures and information system interactions via graphic user interfaces through such techniques such as screen scraping and virtual click/virtual text entry.

In the literature, it is often stated that RPA software serves to automate procedures and integration of disparate information systems by imitating human behavior. The scope of such imitation is rarely defined beyond a specification of RPA capabilities for interacting with information systems through graphic user interfaces designed for use by humans. Unlike efforts to simulate human behavior with chatbots (e.g., delaying chatbot responses to simulate human response times (Gnewuch et al., 2018)), RPA users seem less concerned with simulating human behavior than automating procedures using the graphic user interfaces designed for human users. ***This dissertation research explores an aspect of how RPA can truly simulate aspects of human behavior, by evaluating the potential for RPA to simulate the latency and information degradation of sociotechnical systems.***

RPA is often coupled with artificial intelligence or machine learning technology, yielding “smart RPA” which provides two main benefits. First, AI/machine learning techniques can empower RPA to adapt to changes in the presentation of information in a graphic user interface. If the same information is presented, but the format has changed, a human user can be expected to adapt relatively easily. RPA using rigid screen scraping techniques would need to be adjusted for the new interface format. A second benefit of AI-enhance RPA is an ability to autonomously develop new, increasingly efficient or effective processes for a task.

This dissertation research does not address smart RPA, instead focusing on how workflows may be designed using basic RPA functions to simulate the latency and information degradation expected of target sociotechnical systems. ***It is clear, however, that future research is necessary to determine how to balance the opportunities and risks associated with leveraging smart RPA for training environments.*** While an RPA ability to adapt to changes in user interfaces may be desirable (a human user would not be thrown off by a minor format change), some AI-driven process modifications may not be. The autonomous development of optimized processes would likely run counter to simulating the dynamics of real-world sociotechnical systems. This research will provide a foundation for future investigations regarding the implications of smart RPA for simulating the dynamics of sociotechnical systems.

2. Scoping Sim-IS Environments

The term "sim-IS environments," broadly interpreted, can refer to any simulation-supported environment consisting of both simulations and live information systems. For this research, however, the term is used to refer to those environments which include constructive simulations which provide pucksters "ground truth" information (through graphic user interfaces) which is used to populate real-world information systems (through graphic user interfaces) used by the simulation participants/training audience. Such sim-IS environments are often used to support staff training and wargaming in military or emergency management communities, though they may also support CSE staged worlds and management training in other domains as well.

Sim-C2 environments are a subset of sim-IS environments, as many C2 systems (e.g., C2PC) can be populated through either standardized protocols (e.g., over-the-horizon

[OTH] Gold) or the C2 system's graphic user interface. In these instances, there may be value in leveraging both a sim-C2 protocol and RPA-based information exchange methods for representation of machine-to-machine and human-to-machine information exchange.

The subset of sim-IS environments used for this dissertation research includes military constructive simulations used for staff training and web-based logistics information systems used to convey equipment, supply, and personnel statuses among distributed military units. Logistics information systems are often integrated with constructive training simulations via pucksters for staff training and wargaming evolutions. ***Focusing on this subset of sim-IS environments will yield tangible benefits for military training organizations developing sim-IS environments. Focusing on this problem space also facilitates a clear extension of lessons to other military functions (e.g., intelligence, communications) and to the development of sim-IS environments for logistics management communities beyond the military.***

3. Assumptions

Training designers are assumed to have the requisite resources to acquire and/or develop models of BP-IS integration, either as they currently exist or as anticipated in order to guide the design of sim-IS environments. They are similarly assumed to be capable of securing adequate descriptions of information delay/degradation (current or anticipated) in target sociotechnical systems to guide the design of sim-IS environments. While these models are expected to coevolve with the sim-IS environment over time, initial BP-IS models are assumed to be available or able to be developed to serve as the referent for the initial sim-IS environment.

Assumptions are also made regarding the specific nature of information delay and degradation to be represented by the artifact. This research does not include field studies and data collection/analysis which would be necessary for identifying the specific attributes of information delay and degradation as it propagates through the target sociotechnical processes. Instead, the descriptions generated in coordination with Marine Corps training designers and logisticians for information delay (e.g., time between supply status measurement and status reporting) and degradation (e.g., human error in entering values) are assumed to be sufficient for the purposes of this research.

Evaluating the training effectiveness of the sim-IS training environments to be supported by this new sim-IS information exchange approach is beyond the scope of this research. Extensive research in the fields of naturalistic decision-making abilities, situated cognition, and human-machine teaming has demonstrated the importance of exposing trainees to the true capabilities and limitations of complex information systems in context. This research will not conduct a training effectiveness evaluation of the prototype. It is assumed that simulating the target dynamics of the sociotechnical system for trainees will enhance their understanding of the capabilities and limitations of the information systems in the context of the existing or anticipated business processes. This research is focused on evaluating the technical feasibility of leveraging RPA technology to support automated sim-IS information exchange and simulation of target BP-IS dynamics in support of training requirements specified by the research sponsor. This research is focused on validation of an RPA-based sim-IS information exchange approach through validation of the prototypes. Validation (and evaluation) of the training effectiveness of RPA-based sim-IS environments in supporting the development of appropriate mental models of envisioned integrated business processes should be explored in future research.

C. RESEARCH METHODOLOGY

This research leverages design science and mixed-methods research approaches. Design science research derives from Herbert Simon’s conception of the science of the artificial, and has been developed in the information systems community to support the study of IS artifacts (Pries-Heje et al., 2008; Simon, 1996). The design science research (DSR) approach has been employed to support exploration of live, virtual, constructive (LVC) simulation interoperability (Kim, 2015), an agent-based simulation approach to evaluating IS-supported crisis response coordination (Gonzalez, 2010), and the design of experiential learning environments that include tightly coupled simulations and business information systems (e.g., ERP) (Loffler et al., 2019; Nisula, 2019). Here, design science methods have guided the development and validation of a new, RPA-based approach for a loose coupling of simulations and information systems.

1. Design Science Research

Simon asserted that while natural science is focused on understanding "how things are," artificial science focuses on "how things ought to be" (Simon, 1996). The design science research methodology evolved over the 1990s and 2000s to supports this focus, with an emphasis on guiding researchers' identification of important, relevant problems, demonstration of the novelty of their solution, and evaluation or validation of the solution's utility in context (Hevner et al., 2004; March & Smith, 1995; Peffers et al., 2006). Design science research provides an epistemological foundation for research in the artificial or "design sciences" (see Figure 29), including modeling and simulation.

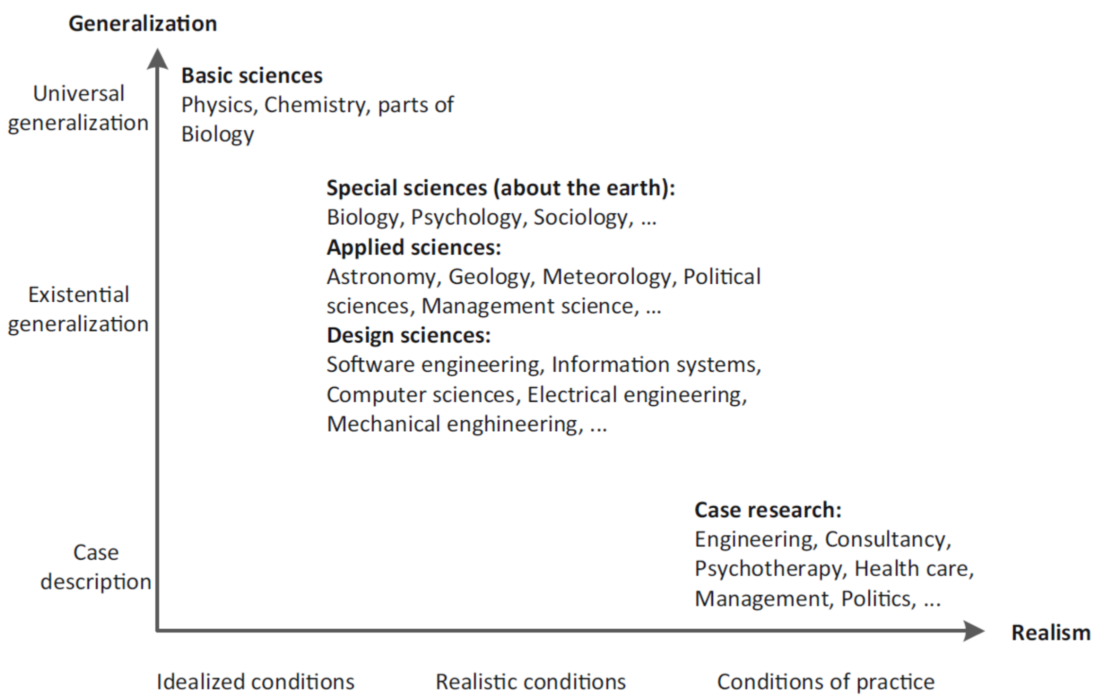


Figure 29. Design Science Relative to Other Sciences. Source: Wieringa (2014).

What distinguishes design science research from routine design of information systems is the knowledge contribution for the community which the design is intended to support. For design science research to be considered a meaningful contribution, the researcher must demonstrate the importance/relevance of the problem and the novelty of the solution presented, in addition to conducting an evaluation and/or validation regarding the utility of the proposed solution (Hevner et al., 2004).

> Routine design is the application of existing knowledge to organizational problems, such as constructing a financial or marketing information system using best practice artifacts (constructs, models, methods, and instantiations) existing in the knowledge base. On the other hand, design-science research addresses important unsolved problems in unique or innovative ways or solved problems in more effective or efficient ways. The key differentiator between routine design and design research is the clear identification of a contribution to the archival knowledge base of foundations and methodologies. (Hevner et al., 2004, p. 81)

The design science research methodology includes: 1) identifying a relevant problem within a community and a gap in existing capabilities/knowledge for resolving the problem, 2) defining objectives for a solution, 3) designing and developing a solution, 4) demonstrating the solution/artifacts , 5) evaluating or validating the associated artifact(s), and 6) communicating the contribution to the community (Hevner et al., 2004; Peffers et al., 2006). Hevner et al. (2004) defined four types of artifacts which may be developed and evaluated in design science research: construct, model, method, and instantiation.

While the DSR methodology, the four types of DSR artifacts, and an appreciation of the origins of DSR in the work of Herbert Simon are all fairly common throughout DSR literature, the distinction between artifact evaluation and validation in DSR is less clearly established. Evaluation and validation of design science artifacts is a critical step in the DSR in information systems is often focused on evaluation of design artifacts. Hevner et al. (2004) presented the evaluation methods specified in Table 3, without discussion of their potential utility in support of artifact validation.

Table 3. Design Science Evaluation Methods. Source: Hevner et al. (2004).

Design Evaluation Methods	
1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation - Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

The term "evaluation" is used as a general term in much DSR literature to refer to both evaluation and validation (Hevner et al., 2004; Peffers et al., 2006; Pries-Heje et al., 2008; Venable, 2006). Venable (2006) classifies DSR evaluation approaches as consisting of either artificial or naturalistic evaluation, a classification which is useful but not sufficient for contrasting validation and evaluation approaches. While both validation and evaluation can be conducted in artificial environments, validation of DSR artifacts must be conducted in artificial environments. This is because validation is conducted before an artifact is implemented, while evaluation is conducted after implementation (Wieringa & Morali, 2012). Validation of artifacts often includes the use of modeling, simulation, and experimental environments where researchers exercise the artifact in the envisioned operating environment to observe its performance in context (Gonzalez, 2009; Wieringa, 2014).

> Validation and evaluation are different research goals that require different research approaches. The goal of validation is to predict how an artifact will interact with its context, without actually observing an implemented artifact in a real-world context…The goal of evaluation research, by contrast, is to investigate how implemented artifacts interact with their real-world context. (Wieringa, 2014, p. 31)

The research goal of validation is to "justify that [a proposed solution] would contribute to stakeholder goals when implemented in the problem context" (Wieringa, 2014, p. 59). Design validation is a particularly apt research approach in situations where the implementation of experimental design science artifacts is impractical due to risk (e.g., air traffic control, space operations) or limited availability of the operating environment (e.g., military or emergency management operations) (Gonzalez, 2009). It is also important to understand how validation of design science artifacts relates to the validation of simulations in the field of modeling and simulation.

In the field of modeling and simulations, the verification and validation process is essential for ensuring M&S tools are properly aligned with the situations which they are intended to support. Verification has been defined as "ensuring that the computer program of the computerized model and its implementation are correct" (Sargent, 2007, p. 124) and is commonly referred to as "modeling a thing right." Validation is defined as "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (Sargent, 2007, p. 124) and is commonly known as "modeling the right thing." In other words, "validation is substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the study objectives" (Balci, 1994, p. 217). This simulation validation is often conducted to ensure a simulation is acceptable for research or training environment requirements, often in support of simulation development or acquisition. In design science research, where an artifact serves as a model of a novel design solution, M&S validation of the artifact relative to the research question(s) can support DSR validation of the underlying design being presented (Gonzalez, 2009).

This research includes development and validation of two design science research artifacts: a prototype RPA-based sim-IS information exchange mechanism (an instantiation artifact) and recommendations for a sim-IS DSEEP overlay (a method artifact). The research questions specified previously were explored through the design, development, and DSR validation of these artifacts. The RPA-based sim-IS architecture was validated through experimental simulation in a controlled laboratory and demonstration in a field environment for domain SMEs. The former environment supported quantitative validation of performance while the latter supported qualitative validation of the artifact's anticipated utility by the research sponsor. The descriptive, informed argument method is leveraged for validation of the DSEEP overlay recommendations, supported by the experimental simulation results and a survey conducted within the Federal RPA Community of Practice regarding RPA conceptual modeling practices.

2. Research Hypotheses

The development and validation of the instantiation and method artifacts supported testing of the research hypotheses associated with the primary question and subsidiary research questions.

Hypotheses associated with the primary research question:

Ho 1: An RPA-based, outside-in approach to automated sim-IS information exchange produces a representation that simulates *temporal dynamics* of information exchange of a target sociotechnical system that is not sufficient to support staff training.

HA 1: An RPA-based, outside-in approach to automated sim-IS information exchange produces a representation that simulates the *temporal dynamics* of information exchange of a target sociotechnical system sufficient to support staff training.

Ho 2: An RPA-based, outside-in approach to automated sim-IS information exchange produces a representation that simulates the *degradation* of information exchanged in a target sociotechnical system that is not sufficient to support staff training.

HA 2: An RPA-based, outside-in approach to automated sim-IS information exchange produces a representation that simulates the *degradation* of information exchanged in a target sociotechnical system sufficient to support staff training.

Hypotheses associated with (subsidiary) question 2:

Ho 3: An RPA-based, outside-in approach to automated sim-IS information exchange produces an exchange of ground truth simulation information to a target information system with *insufficient* accuracy, precision, or timeliness to support staff training.

HA 3: An RPA-based, outside-in approach to automated sim-IS information exchange produces an exchange of ground truth simulation information to a target information system with *sufficient* accuracy, precision, and timeliness to support staff training.

The first three hypotheses (H1, H2, and H3) are provided here in general form. Specific requirements for a "sufficient" representation of latency and information degradation within the target sociotechnical systems to be simulated (e.g., probability distributions for latency to be represented) were identified in coordination with the primary research sponsor and are discussed in Chapter V. Specific requirements for a sufficient level of accuracy, precision, and timeliness in exchange of ground truth information between simulations and information systems are also addressed in Chapter V.

3. Instantiation Artifact

The RPA-based sim-IS middleware prototypes were built using UiPath, a commercial-off-the-shelf RPA software used within civilian and DOD organizations to automate back-office work procedures including HitL ISs. The sim-IS information exchange prototypes automate the exchange of information between constructive simulations (MTWS and Joint Conflict and Tactical Simulation [JCATS]) and a Marine Corps logistics information system (CLC2S). The simulations, IS, and information to exchanged were identified in coordination with the primary sponsors of this research; MCLOG.

Generic integrated business processes were defined, in coordination with MCLOG, which loosely reflect Marine Corps logistics processes. These business process models served as the referent for the design and development of the RPA workflows for sim-IS information exchange. They also included specification of BP-IS integration dynamics (information delay and degradation [e.g., human error]) to be simulated by the RPA-based sim-IS information exchange prototypes. These BP-IS dynamics include the kinds of issues which necessitate the Marine Corps Training and Readiness task "Validate Support Requests," which challenges logistics staffs to apply critical analysis to all logistics support requests and status reports. A thorough description of the design of the modular RPA-based sim-IS information exchange architecture is provided in Chapter IV.

4. Method Artifact

Following the development and validation of the RPA-based sim-IS information exchange prototypes, recommendations regarding a methodology for the design and development of RPA-based sim-IS environments was developed, addressing the third subsidiary research question. This was conducted through a review of existing BP-IS and RPA modeling approaches, current DSEEP guidelines, and identification of BP-IS attributes which may impact decision-making. This research provides the groundwork for development of a DSEEP overlay for the design and development of sim-IS environments, with an emphasis on supporting an RPA-based approach to sim-IS information exchange. These recommendations are intended to support the design of RPA workflows for sim-IS information exchange, but also support other sim-IS information exchange approaches. Issues to addressed include:

- identifying attributes of BP-IS dynamics to be represented in a conceptual model for sim-IS environments (e.g., information latency, human error),

- identifying a methodology for identifying (or building) BP-IS models which serve as referents for the design of sim-IS environment conceptual models, and which capture the nature of information delay and degradation to be represented,

- identifying how such a sim-IS CM may be employed to evaluate the composability of a simulation and IS for the target BP-IS process(es),

- defining a data exchange model for RPA-based sim-IS interoperability which captures BP-IS dynamics for HitL ISs, and

- identifying how such a sim-IS CM may be employed to guide the development of RPA workflows for sim-IS information exchange.

5. Verification and Validation Methods

Verification and validation of the RPA-based sim-IS information exchange approach is supported by quantitative and qualitative methods applied to prototype in artificial environments, including simulation in a controlled environment and demonstration of the prototype with the MCLOG M&S staff. Simulations was conducted in the Naval Postgraduate School (NPS) Modeling, Virtual Environments, and Simulation (MOVES) Institute's training and simulations lab throughout the development of the prototype. These simulations supported iterated development of the prototypes and a final, quantitative verification of the tool for automating sim-IS information exchange and simulating the requisite BP-IS dynamics with the delay and degradation of information exchanged. This verification process is discussed in Chapter V. A demonstration of the prototypes was conducted with MCLOG M&S staff to support qualitative validation of the approach, with an emphasis on the utility, functionality, and fit of the prototype in support of Marine Corps sim-IS staff training environments.

The measures and thresholds used in the quantitative and qualitative evaluation of the artifacts were determined in coordination with the research sponsor and in accordance with design science research guidelines. Rigorous verification and validation of both artifacts necessitates identification of not just appropriate verification and validation methods but also appropriate metrics to demonstrate the "utility, quality, and efficacy" of the artifacts (Hevner et al., 2004, p. 85). Recall that this research assumes the sim-IS environment designer can identify the requisite STS dynamics to be simulated for the development of desired NDM abilities for a given integrated business process. This research does not address whether the STS dynamics specified by the training designer are the correct dynamics to simulate, only whether the RPA-based sim-IS information exchange architecture adequately simulates the STS dynamics as they are specified. As discussed in Chapter V, multiple sets of STS dynamic distributions are tested to ensure the instantiation artifact is tunable to support the simulation of different STS dynamics as sim-IS environments and the supported integrated business processes coevolve.

In some instances, such as evaluating the artifact's ability to automate information exchange, the performance of an artifact can be proven simply by demonstration that it works: "proof by construction" (Hevner et al., 2004, p. 84). Other research questions, especially those addressing the ability of the artifacts to simulate dynamics such as latency and content degradation in information exchange, require more thorough quantitative and qualitative verification and validation methods. Table 4 provides an overview of general methods and metrics employed for the validation of each artifact in support of the specified research questions. A more thorough review of the verification and validation process is presented in Chapters V and VI.

Table 4. Evaluation Methods and Metrics by Artifact and Research Question.

Artifact	Research Question	Metrics	Validation Method(s)
RPA-based middleware Type of Artifact: Instantiation	1) Simulating temporal dynamics for sim-IS information exchange	- Accuracy in aligning the latency and timeliness of information exchange in RPA schedules with the target distributions	- simulation (NPS lab) (quantitative analysis)
	1) Simulating information content degradation for sim-IS information exchange	- Accuracy in aligning the introduction of information content degradation with the target content degradation probabilities	- simulation (NPS lab) (quantitative analysis)
	1) Simulating STS dynamics in sim-IS information exchange	- Accuracy in simulating sim-IS information exchange in accordance with RPA schedules	- simulation (NPS lab) (quantitative analysis) -demonstration (MCLOG) (qualitative analysis)
	2) Achieving sim-IS information exchange through RPA	- Automation of sim-IS information exchange (proof by construction) - Functionality (accuracy, precision, and timeliness of sim-IS information exchange)	- simulation (NPS lab) (proof by construction / quantitative analysis) -demonstration (MCLOG) (qualitative analysis)
	3) Achieving a MOSA design for sim-IS information exchange with RPA	- Utility (steps required to configure RPA modules for different sim-IS environments [e.g., MTWS-CLC2S to JCATS-CLC2S])	- simulation (NPS lab) - informed argument
Sim-IS CM Development Guidelines (DSEEP Overlay Recommendations) Type of Artifact: Method (Process)	4) Designing sim-IS CMs to represent BP-IS processes and STS dynamics	- Functionality, utility, completeness	- simulation (NPS lab) - informed argument (qualitative analysis)

IV. RPA-BASED SIM-IS ARCHITECTURE

This dissertation presents an RPA-based sim-IS architecture which supports the simulation of sociotechnical system dynamics across varied combinations of simulations and HitL information systems. Unlike existing approaches to sim-IS interoperability, which rely upon either human pucksters or simulation/information system engineering changes for dynamic information exchange, this architecture also supports automated data exchange with low overhead, modular and reusable components. The RPA-based sim-IS architecture, illustrated in Figure 30, can be described in terms of the RPA middleware itself and the sim-IS StartEx data generation process which facilitates the dynamic sim-IS exchange. A more granular conceptual models for the RPA-based sim-IS architecture is provided in Appendix C, along with conceptual models for the RPA modules developed in support of this research.

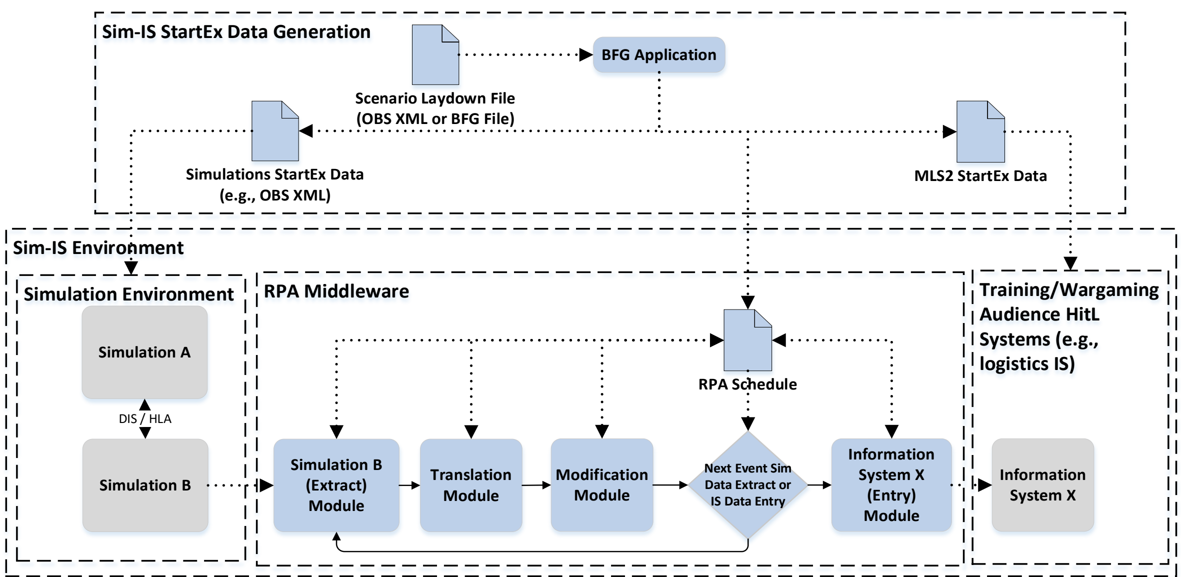


Figure 30. Sim-IS RPA Architecture.

Sim-IS StartEx data generation includes the processes, tools, and standards necessary for generating synchronized scenario data for populating all simulations and information systems included in a sim-IS environment. While there are situations where slight disparities between simulation and information system starting data may be desirable, it is generally important to begin exercises with synchronized data across all simulations and information systems. The sim-IS StartEx data generation process requires the alignment of both scenario data and system parametric data across simulations and information systems. Scenario data includes such information as unit hierarchies and naming conventions, unique identifiers for simulated personnel (e.g., names, numeric identifiers) and equipment (e.g., serial numbers), and the assignment of equipment and supplies to units. The alignment of parametric data includes the mapping of supply item units of measure (e.g., counted as individual items or cases) and ensuring unique identifiers used in simulations' parametric data (e.g., national stock number, enumeration) refer to equivalent equipment or supply classes.

The RPA middleware supports the dynamic synchronization of simulations and information systems in accordance with the sociotechnical system dynamics (e.g., latency, timeliness, errors) required for a given sim-IS environment. The RPA middleware consists of four types of RPA modules: RPA simulation data extraction modules, RPA information system data entry modules, a data translation module, and a data modification module. These modules are supported by sim-IS data transformation components which guide the translation of simulation data for entry into information systems and the modification of translated data for simulation of data degradation across the sociotechnical system. An RPA Schedule file fulfills a function similar to a discrete event simulation schedule, scheduling all sim-IS data exchange events in accordance with target latency and timeliness distributions.

A. SIM-IS ENVIRONMENT ARCHITECTURE

The RPA-based sim-IS environment architecture consists of a modular open systems approach (MOSA) to support RPA module reuse and modernization. The RPA workflow modules for extraction of information from simulations and entry of data into information systems are reusable. Each module is intended to be built once for a specific simulation or information system, respectively, to support information exchange with different combinations of simulations and information systems, as depicted in Figure 31. Modifications may be required for IS data entry modules as processes for entering data into information systems evolve. Modifications may be required for simulation data extract modules in order to better align with the target processes.

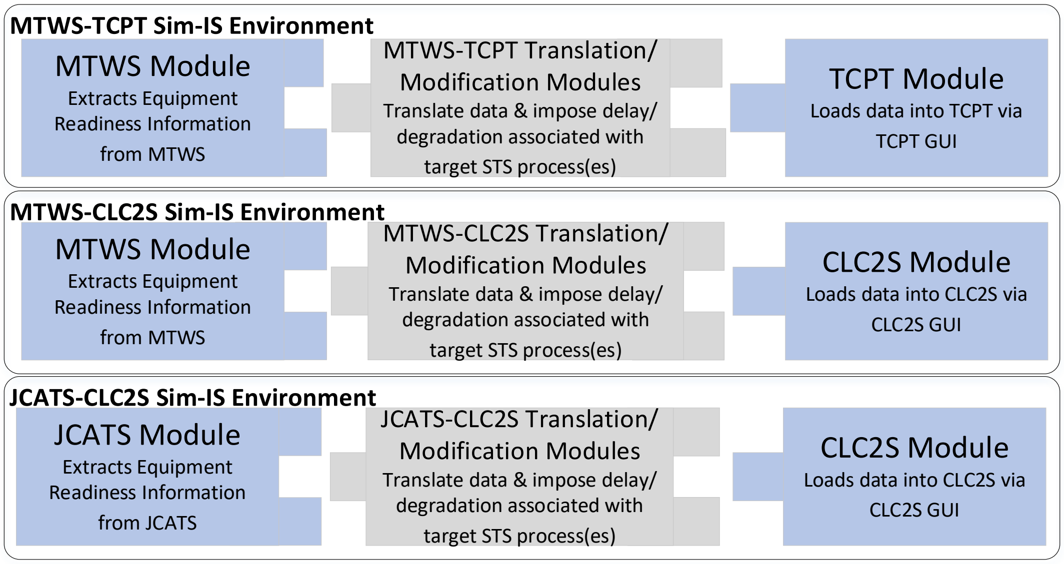


Figure 31. Modular Design for RPA Modules in Sim-IS Data Exchange.

Translation and modification modules couple a specific set of simulation extraction and IS data entry modules. The translation and modification modules are unique to simulation/ IS(s) they support and to the integrated business process(es) being simulated in the sim-IS environment. Translation modules support the mapping of simulation and information system data elements (e.g., entity readiness levels and supply units of measure). The modification module applies the specified STS dynamics to be simulated in the sim-IS environment (e.g., timeliness, latency, and human data entry errors), by modifying extracted supply or entity readiness values and guiding the scheduling of future events. While these modules are reusable (MTWS and CLC2S information exchange may be required for many scenarios), they must be tailored for the sociotechnical dynamics to be simulated for a given sim-IS environment.

1. RPA Schedule and Sim-IS Data Transformation Components

The sim-IS RPA schedule supports scheduling of sim-IS events while the rest of the sim-IS Data Transformation components support the translation of extracted simulation data to the equivalent information system values and simulation of sociotechnical system dynamics. The data transformation components, identified in Figure 32, are developed at various times in the sim-IS environment design and development using standardized templates. Several of these data transformation components (supply types table, entity readiness table, and the STS Dynamics tables) also include unique exercise design considerations to be addressed at different stages of sim-IS environment design.

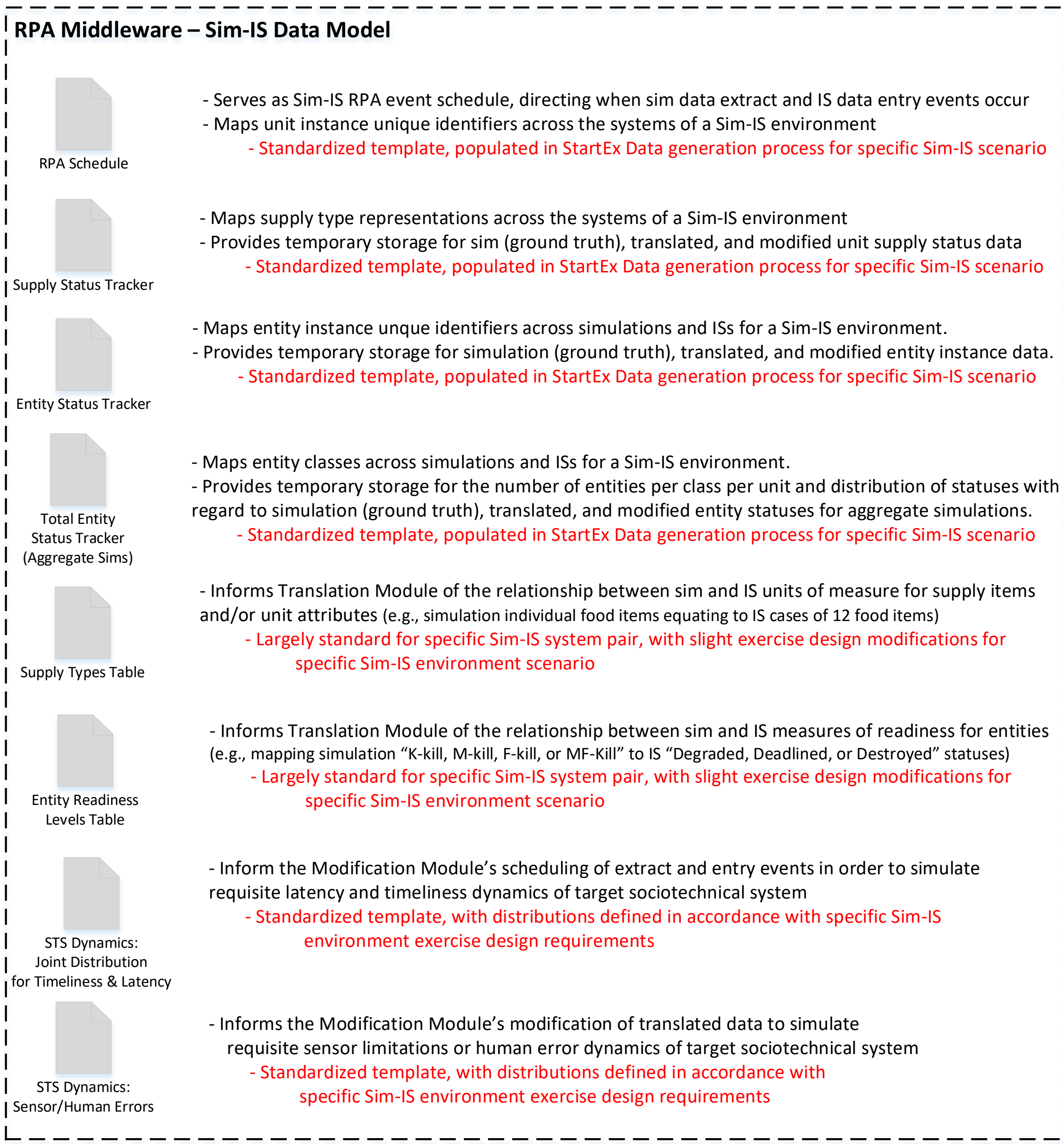


Figure 32. Sim-IS Data Transformation Components.

Figure 33 illustrates how the RPA schedule specifies the time of simulation extract and information system entry module execution and for which units, supplies, and entities. This schedule is initialized in the sim-IS StartEx data generation process and is updated throughout the sim-IS environment execution, like a discrete event simulation schedule. The translation and modification modules always occur immediately after a simulation extract module is executed.

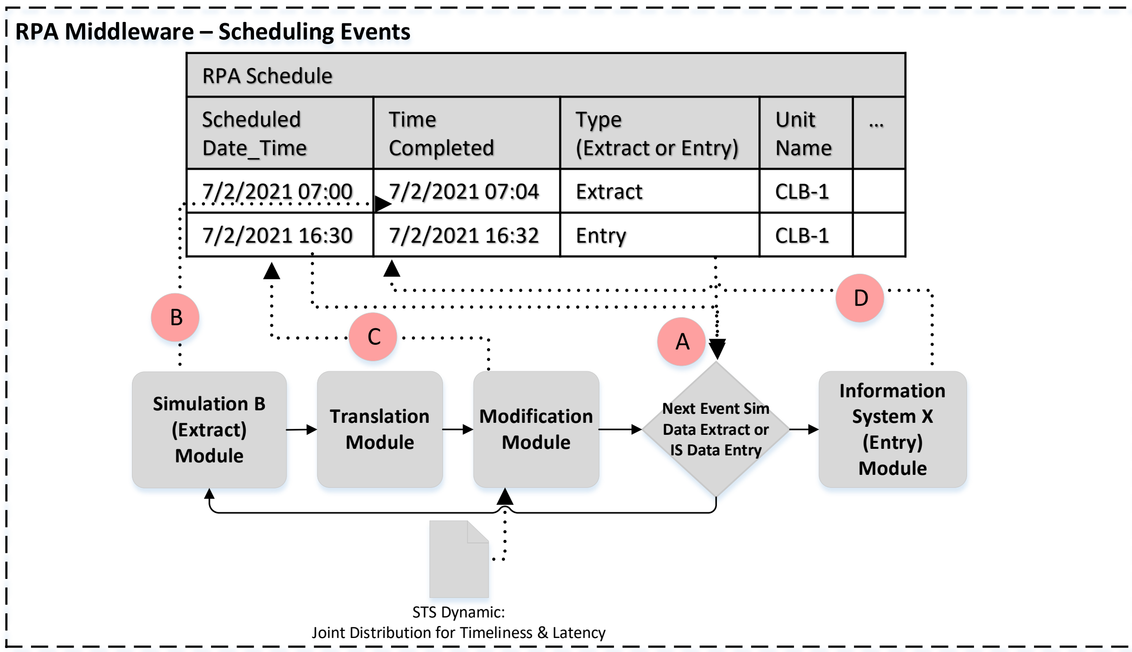


Figure 33. RPA Schedule Role in Event Scheduling and Execution.

The interaction of RPA modules with the RPA schedule for scheduling and execution of events is as follows (referencing Figure 33):

- Step A: RPA Schedule identifies the next scheduled event to occur and determines the event type (sim data extract or IS data entry)

- Step B: Sim data extract module updates the “time completed” field for sim data extract events following execution at the scheduled time

- Step C: Modification module schedules the next extract and entry events (likely for the next day) for the given unit in accordance with the timeliness and latency distributions specified for the sim-IS environment

- Step D: IS data entry module updates the "time completed" field for IS data entry events following execution at the scheduled time

The rest of the data transformation components can be divided into three groups: status tracker files, translation files, and modification files. The status tracker files (supply status tracker, entity status tracker, and total entity status tracker) support temporary storage of supply and entity readiness values. The status tracker files, along with the RPA Schedule, also support the mapping of unique identifiers for units and entities across the simulation and IS.

The translation files include the supply status table (Table 5) and the entity readiness levels table (Table 6). These tables associate potential simulation entity readiness values and supply item units of issue from with their equivalent values in the IS. While the potential values for each system may be static, the way they are mapped to between the simulation and IS constitutes a sim-IS design choice. For entity readiness levels, for example a sim-IS environment designer may choose to make an environment more less challenging by assigning "Wounded (Priority)" status in MTWS with "Wounded in Action (Walking)" or "Wounded in Action (Evacuation)."

Table 5. Example Supply Status Table.

Supply Item Nomenclature		Units of Issue		Conversion Rate
MTWS	CLC2S	MTWS	CLC2S	MTWS/CLC2S
Fuel	JP8	Gal	Gal	1
Rations	Meal Ready to Eat	Each	Case	1/12
.50Cal	.50Cal	Each	Each	1
Water	Water	Gal	Gal	1

Table 6. Example Entity Readiness Levels Table

Entity Type	MTWS Statuses	CLC2S Statuses
Personnel	Healthy	Full Duty
	Wounded (Routine)	Wounded in Action (Walking)
	Wounded (Priority)	Wounded in Action (Evacuation)
	Wounded (Urgent)	Wounded in Action (Evacuation)
	KIA	Killed in Action
Equipment	Operational	Available
	M-Kill	Deadlined
	F-Kill	Deadlined
	MF-Kill	Deadlined
	K-Kill	Destroyed

The modification files include two for specifying the desired latency distribution for the exchange of information and timeliness distribution for time of data entry in the IS. There is also a "Sensor/Human Errors" table for specifying the desired frequency for simulating different types measurement and data entry errors. Unlike the latency and timeliness distributions, the simulated errors table is not read into the modification module, but instead informs the design and parameters for the errors built into the modification module.

2. Unique Identifiers for Sim-IS Environment Units and Entities

Exchanging updates regarding particular unit or entity statuses across multiple systems requires the use of unique identifiers to ensure the update is occurring between the appropriate units/entities across the systems. For distributed simulations, the HLA architecture addresses this through unique unit/entity identification numbers called "LVC IDs." In HLA environments, simulations communicate changes in entity statuses by referring to the unit/entity LVC ID. The LVC IDs used for a given HLA federation are specified in the OBS XML file which is used to populate all HLA federates.

Sim-IS environments cannot use this approach without imposing engineering changes on information systems (and simulations) which are not built to use LVC IDs. Rather than impose a set of common unit/entity IDs on the participating simulation and IS, the designer of simulation data extraction and IS data entry modules must identify unique identifiers for units and entities within the existing user interface. These unique identifiers may be serial numbers, names, or even the position of a unit or entity within a unit structure. The unique identifiers must also be present in the in the user interface for the associated simulation or IS, in such a way that they are associated with the entity or unit's readiness and/or supply statuses.

Even when simulations are built to operate within an HLA environment, LVC IDs are often not included as references in a simulation's GUI. For JCATS, unique JCATS ID numbers are assigned to units and entities. JCATS IDs are used to request unit or entity statuses in the JCATS Web Bridge reports. In these situations, the simulation data extract module developer must identify how GUI unique identifiers can be specified in the RPA transformation components.

The mapping of JCATS IDs to the corresponding unit/entity unique identifiers used by IS entry modules is through the LVC ID. The JCATS Web Bridge produces a report called an "orbat report," which specifies both the LVC ID and JCATS ID for every unit and entity in a given scenario. This allows the JCATS extract module to extract the JCATS ID and LVC ID pairs, which facilitates mapping JCATS IDs to the equivalent unit and entity unique identifiers for the associated IS in the sim-IS environment. For JCATS, the JCATS ID and IS unique identifiers are mapped for each unit/entity instance via a common LVC ID which is associated with IS unit/entity unique identifiers during the sim-IS StartEx generation process. That process, and the role of the Batch File Generator application in automating the synchronization of sim-IS StartEx files, is discussed in the sim-IS StartEx Data Generation section.

For MTWS, there is no requirement for an additional "Sim_Identifier" field, as the unit name is the only unique identifier for each unit. As an aggregate simulation, individual entities are not tracked and therefore require no unique ID. The only requirement for ensuring unit statuses can be conveyed appropriately for paired IS entry modules is to ensure unit names are unique. Similarly, equipment, personnel, and supply class names must be unique.

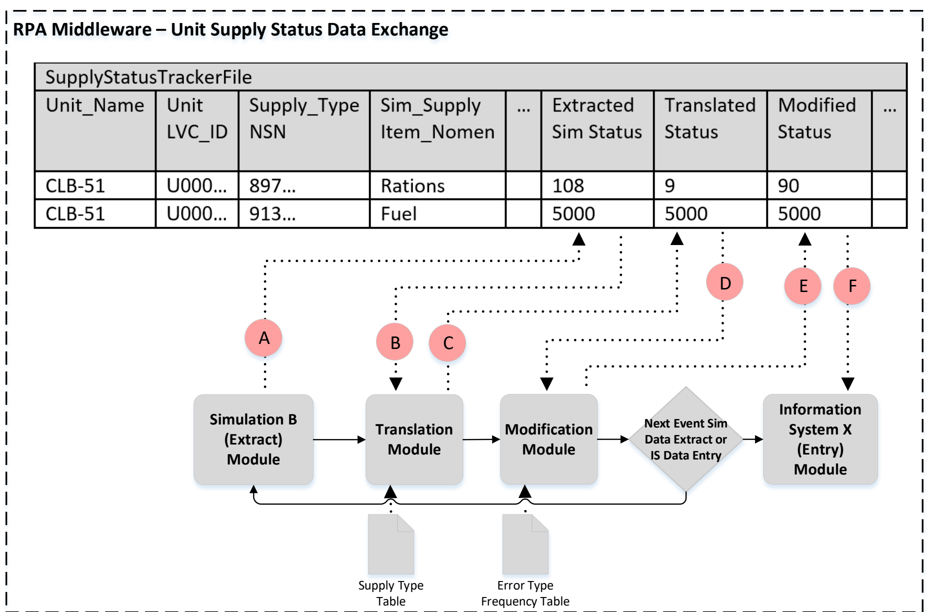
The modification module for MTWS includes a process whereby the aggregate entity statuses specified in the "total entity statuses file" are used to adjust the statuses for CLC2S entities so that the aggregate CLC2S entity statuses align with those of the corresponding MTWS unit.

3. Sim-to-IS Data Translation

The sim-IS data translation module supports the syntactic and semantic levels of interoperability for sim-IS data exchange. Syntactic interoperability includes ensuring sim-IS data is conveyed in a compatible format across the systems. This can include converting a floating point value to an integer, or a number to a category value based on the corresponding ranges of numbers per category. The translation module is designed to ensure the appropriate translation of values between a given simulation and IS pair. Translation module design is informed by the documentation for corresponding simulation and IS RPA extract/entry modules, which must capture details regarding syntax of data output from the simulation and requirements of input into the IS GUI.

Semantic interoperability includes ensuring the meaning of the values exchanged is aligned across the sim-IS systems. One example of a semantic interoperability challenge for sim-IS environments is aligning the representation of unit supply levels. Some combat simulations (e.g., MTWS) do not distinguish between unit supplies held in unit bulk stores and the supplies which have been distributed to individual vehicles (e.g., fuel, ammunition) and personnel (e.g., food, water, ammunition). This poses a semantics challenge because logistics information systems are often used to report unit bulk stores levels, with an implicit assumption that additional fuel exists within vehicle tanks and additional rations and water exist within personnel packs. In these instances, the translation module should be designed to adjust reported supply levels accordingly. For fuel reporting, this could be accomplished by calculating the fuel capacity of all vehicles within a given unit, selecting an assumed level of fuel in unit vehicles (say 80%) and then decrementing the total stores accordingly.

Another sim-IS semantic interoperability consideration addressed in the translation module is aligning supply item units of issue. Translation of supply quantities and entity readiness levels across simulations and information systems requires a system for mapping unit instances, entity instances, supply type unique identifiers, supply type quantities (unit of issue), and entity status levels across simulations and information systems. The mapping of unit/entity instances and supply types is addressed by through the sim-IS unique identifiers addressed earlier. The mapping of supply type quantities and entity status levels may be straightforward, if the number and type of readiness values are similar. In such instances, the translation of simulation to IS statuses may only require a standard conversion rate, as seen in Table 5. The process for sim-IS data translation and modification is illustrated in Figure 34.



The role of this file is to temporarily store the extracted, translated, and modified simulation ground truth supply status data for entry in the target information system(s).

Figure 34. Supply Status Tracker File

- Step A: Extract Module extracts supply quantity (or attribute) from Simulation B's GUI and enters it into the supply status tracker

- Steps B-C: Translation Module reads the extracted supply quantity, translates it into the equivalent quantity (or attribute status) for the information system using the data model, and enters translated values into the supply status tracker

- Steps D-E: Modification Module reads the translated supply quantity (or attribute status), imposes errors in accordance with the error distributions specified for the sim-IS environment, and enters the modified value into the supply status tracker

- Step F: Entry Module reads the modified supply status from the supply status tracker and enters it into the information system GUI

In the example provided by Figure 34, the supply item (rations or meals ready to eat [MRE]) is tracked with different units of issue in the simulation and information system. While simulation B tracks food by individual meals, the information system tracks them by the case (12 meals). The translation module identifies this difference when referencing its data model for supply types, and converts the simulation supply quantity (108 MREs) to the equivalent supply quantity for the information system (9 cases). In this example, the modification module has also been designed so that there is some probability of an extra digit being erroneously added to the supply value being reported.

When the paired simulation and IS leverage different number or types of supply quantity and equipment/personnel readiness levels, as seen with Table 5 and Table 6, the sim-IS environment designer may encounter a design challenge. If a simulation only presents "K-kill," "M-kill," and "F-kill" entity readiness statuses, for example, the sim-IS and training environment designer(s) must determine how to align these readiness statuses with various statuses which trainees must be prepared to respond to in their information systems (e.g., degraded, deadline, destroyed, missing).

4. Sim-IS Data Modification

The modification of translated simulation data refers to the simulation of sociotechnical system dynamics through the degradation of data reported values (e.g., simulating human errors) and/or the scheduling of simulation extract and information system entry events to simulate data exchange timeliness variability and latency. The sim-IS data modification module simulates the context of data exchange between simulated business processes and associated IS, and in so doing it supports a higher level of interoperability: pragmatic. This includes simulating sociotechnical system dynamics like variability in information latency and in the timeliness of its entry, as well as human/system errors in data entry. It also includes simulating the processes through which simulated units populate the associated IS. Figure 35 illustrates the process for simulating sociotechnical system dynamics addressed in this research.

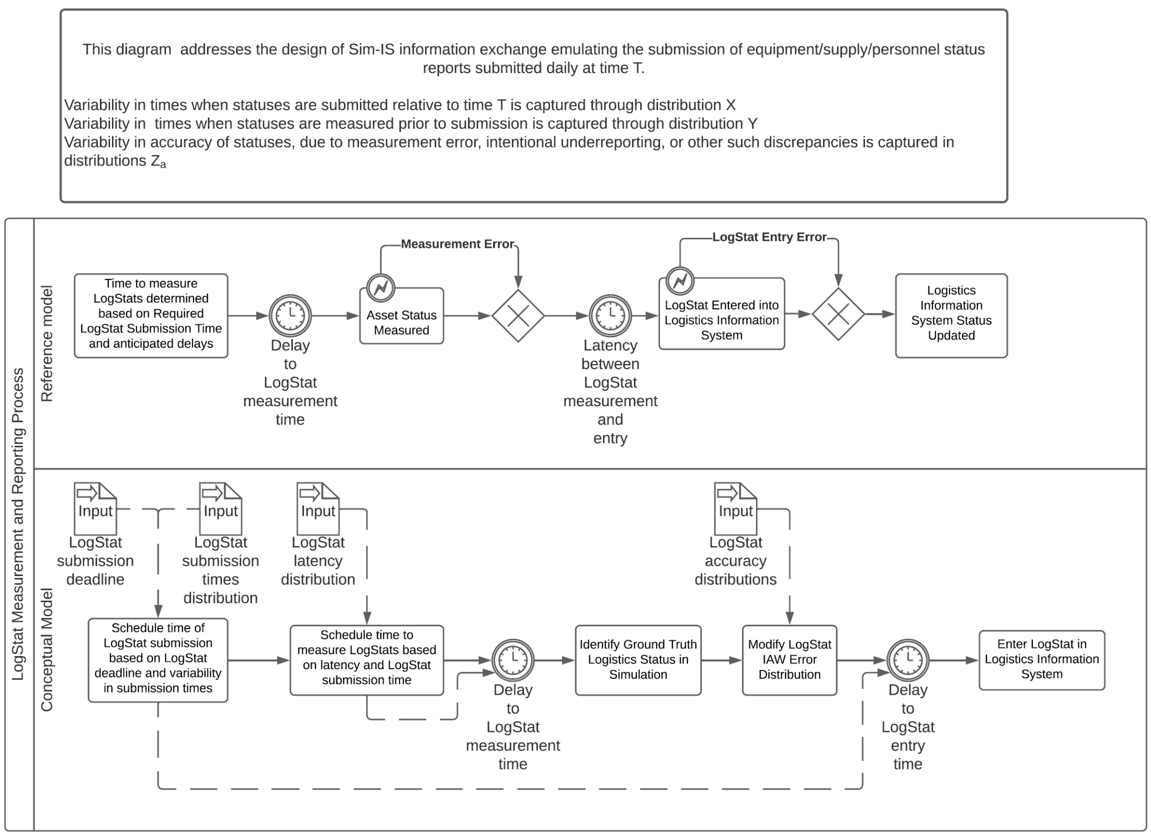


Figure 35. Conceptual Model of Sociotechnical System Dynamics Simulated in a Sim-IS Environment.

a. Simulating STS Dynamics: Information Exchange Latency and Timeliness

The simulation of sociotechnical system dynamics like latency, timeliness, and human error can be addressed separately from the simulation of integrated business processes. While the two are linked (latency, timeliness, and data degradation are directly influenced by the design of integrated business processes and vice versa), the quality and quantity of STS dynamics can be defined separately. For this research, latency refers to the amount of time which elapses between the measurement of a value (e.g., supply status or personnel/equipment readiness level) and when it is reported in the associated IS. Latency distributions define the variability in latency to be simulated in sim-IS exchanges occurring across multiple units and simulation days.

Timeliness refers to variability in the times when data is entered into the appropriate IS relative to the scheduled entry time. For example, if unit logistics statuses (logstat) are due at 1700, it is unlikely that all units will submit their logstats right at 1700. Some units will submit their logstats early and some will be submitted late. Timeliness and latency distributions are specified by the sim-IS designer in the form of a joint distribution table which is save in the RPA middleware directory. The joint distribution table specifies the probability that a pair of latency and timeliness values occur for a given pair of timeliness and latency distributions and are calculated as illustrated in Table 7.

Table 7. Joint Distribution for Latency and Timeliness Dynamics.

Target Timeliness Distribution:			Extract Time												
Timeliness (Hr)	Timeliness Probability (%)	Entry Time	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
-3	0.07	1400	0.0049	0.0077	0.0112	0.0126	0.0168	0.0119	0.0049						
-2	0.08	1500		0.0056	0.0088	0.0128	0.0144	0.0192	0.0136	0.0056					
-1	0.17	1600			0.0119	0.0187	0.0272	0.0306	0.0408	0.0289	0.0119				
0	0.26	1700				0.0182	0.0286	0.0416	0.0468	0.0624	0.0442	0.0182			
1	0.18	1800					0.0126	0.0198	0.0288	0.0324	0.0432	0.0306	0.0126		
2	0.14	1900						0.0098	0.0154	0.0224	0.0252	0.0336	0.0238	0.0098	
3	0.1	2000							0.007	0.011	0.016	0.018	0.024	0.017	0.007
Target Latency Distribution:				Latency Probability (%):					0.07	0.11	0.16	0.18	0.24	0.17	0.07
				Latency (Hr):					6	5	4	3	2	1	0

During the execution of the modification workflow, the RPA platform's pseudo random number generator is used to schedule the times for each unit's sim-IS information extract and entry events for the following day. Each time a particular pair of information extract and entry values is selected, the probability of that pair of values being selected in the future is decremented by 1/n, where n is the number of units for which sim-IS information exchange is executed for a given scenario. This process, illustrated in Figure 36, is similar to sampling without replacement, and ensures adequate simulation of the target timeliness and latency distributions. This is particularly necessary for RPA platforms with pseudo random number generators which are insufficiently random to support simple sampling of the distributions (Appendix E identifies the results of UiPath PRNG randomness testing). This process also facilitates identification of a maximum number of information extract and entry events per hour, which is important for identifying the number of RPA bot instances required for a sim-IS environment and will be discussed in greater detail in Chapter V.

Joint Distribution Values		**Probability of Occurrence**
Extract	Entry	
0800	1400	0.0049
0900	1400	0.0077
1000	1400	0.0112
1100	1400	0.0126
1200	1400	0.0168
1300	1400	0.0119
1400	1400	0.0049
0800	1500	0.0056
...	...	...

→

For n = 1000,

.0049 – 1/n = .0039

Joint Distribution Values		**Probability of Occurrence**
Extract	Entry	
0800	1400	0.0039
0900	1400	0.0077
1000	1400	0.0112
1100	1400	0.0126
1200	1400	0.0168
1300	1400	0.0119
1400	1400	0.0049
0800	1500	0.0056
...	...	...

Figure 36. Sampling from the Joint Distribution for Latency and Timeliness and Adjusting Joint Distributions.

While this research focuses on the simulation of generic temporal distributions, future work may include specification of latency and timeliness distributions that are unique to individual units, simulating unique personalities and processes of different units. After the modification module randomly selects the data entry and extract times, the data extract and data entry events for the associated unit are added to the RPA Schedule file to schedule the next events. In addition to guiding the actions of the RPA middleware, the RPA Schedule also provides exercise control personnel insights into the instances of latency and timeliness variability which their training audience are set to experience. Where sim-IS exchanges are set to occur daily, the RPA schedule can be pulled at the end of the day to observe all of the following day's scheduled extraction and entry events.

b. *Simulating STS Dynamics: Measurement/Reporting Error in Data Exchange*

Another sociotechnical system dynamic simulated through the modification module is measurement or reporting error. This dynamic is also illustrated in Figure 35, with "LogStat Accuracy Distributions" referring to the probabilities specified for having different types of human or system errors manifest in the IS. The examples of data degradation simulated in this research, and the associated probabilities of occurrence, are provided in Table 8. Unlike the latency and timeliness distributions, which are separate files that are read into the RPA module, the "LogStat Accuracy Distributions" actually refers to descriptions of error types and desired probabilities of occurrence which inform the design of the modification module.

Table 8. Examples of Data Degradation Simulated in a Sim-IS Environment and Probabilities of Error Occurrence.

Type of Data	Error Type	Probability of Occurrence
Supply	Failing to enter a new status	4%
	Erroneously adding an extra digit (e.g., "0")	4%
	Rounding the value to the nearest 100	3% (for quantities > 500)
Equipment/ Personnel	Reporting an incorrect status	4%

To impose the data degradation/errors with the specified degree of frequency, the modification module again uses a pseudo random number generator to determine whether to apply one of the specified error types for each of the supply and equipment statuses reported. The modifications are applied to the "translated status" and saved under the "modified status" column. Documentation of both "translated" and "modified" statuses in the supply status tracker and entity readiness tracker files allows exercise control personnel to review the data to be presented to the training audience. If an error is to be simulated, exercise control personnel are then able to observe the training audience response and identify how long it takes the training audience to identify and respond to the discrepancy.

c. *Simulating Integrated Business Processes*

Sim-IS data modification includes more than just simulating STS dynamics like latency and data degradation. The frequency with which unit supply and equipment/ personnel readiness levels are reported (e.g., daily, every 6 hours) and the scheduled times for the submission of the data in associated information systems (e.g., daily at 1700) are two examples of integrated business process attributes which are simulated in the exchange of information across sim-IS environments. The modification module can be adjusted to support different frequencies of sim-IS exchanges and a different LogStat submission time upon which the timeliness variability distribution is applied.

Simulating integrated business processes must also consider how the data source simulated by the simulation(s) influences the IS into which the simulation data should be entered, particularly for sim-IS environments including multiple information systems. It is not uncommon for staffs and decision-makers to encounter conflicting information across multiple ISs. It is important, therefore, for IS operators to understand how these different statuses may result from how each IS is integrated in the broader business process.

For example, multiple Marine Corps logistics ISs support reporting on equipment readiness statuses (e.g., CLC2S, TCPT, Global Combat Support System - Marine Corps [GCSS-MC]). In a deployed environment, the accuracy and level of detail provided in equipment readiness reports may increase as a piece of damaged equipment proceeds from the point of damage to higher levels of maintenance, where more thorough inspections may be conducted. The readiness reporting conducted at different stages during this evacuation are likely to be entered into different information systems at different times. As a result, it may be appropriate for a sim-IS environment to simulate the CLC2S receiving a more timely status update regarding asset availability, while the maintenance IS (GCSS-MC) receives a later, but more accurate and detailed report in the form of a work order being generated. Initial CLC2S reports of an asset being degraded may appear in CLC2S (or TCPT) before a more detailed maintenance report is generated in GCSS-MC showing that the asset is actually deadlined or even destroyed.

Such a disparity in the representation of asset availability is not uncommon in the real world. It may be the result of a temporary, temporal misalignment or reflect a semantic difference in the integrated business processes of different units. In either case, the modification module is where the scheduling of status entries can be scheduled for different ISs, along with simulation of disparities which represent the associated processes. For the scenarios developed in support of this research, all units are assumed to have the same target LogStat report times and frequency in LogStat reporting. Furthermore, all units were assumed to use processes with semantic and pragmatic compatibility. For example, for the JCATS-CLC2S scenario, all units were assumed to report only the supplies held within supply units.

Integrated business processes do not always align, however, and the pursuit of semantic and pragmatic levels of interoperability may include designing a sim-IS environment to simulate the misalignment of unit processes. Simulating these lapses in the real-world semantic and pragmatic interoperability of integrated business processes provides an opportunity to highlight process incompatibilities to be adjusted and standardized. These simulations can also provide decision-makers insight into the capabilities and limitations of their IS in the context of the associated organizational processes.

5. Simulating Lapses in Semantic and Pragmatic Interoperability

For sim-IS environments, achieving an appropriate level of semantic and pragmatic interoperability does not mean achieving semantic and pragmatic interoperability of the data entered into the IS relative to the ground truth information from the simulation. Instead, sim-IS semantic and pragmatic interoperability should simulate the levels of real-world semantic and pragmatic interoperability between the IS and the associated integrated business process.

Achieving semantic and pragmatic interoperability requires an understanding of the processes through which ISs are employed in an operational environment. The degree of interoperability at these levels should simulate that seen in real world integrated business processes. If different units or personnel leverage the same IS in different ways, such differences may result in lapses in semantic and pragmatic interoperability. In these instances, it would be valuable for sim-IS environments to simulate inconsistencies between different integrated business processes, so they can be identified and addressed through process redesign. There are several scenarios where lapses in semantic and pragmatic interoperability can occur in the use of logistics information systems.

For a semantic interoperability example, consider a scenario where two simulated units, Combat Logistics Battalion-1 (CLB-1) and CLB-5, are reporting their fuel supply levels to Combat Logistics Regiment-1 (CLR-1), which is the training audience for a staff training exercise. In this scenario, only CLB-1 has received new vehicle fuel tank sensors which facilitate reporting of fuel levels across all vehicle tanks. If the simulated CLB-1 unit reports its cumulative fuel level while CLB-5 reports only the fuel within its bulk fuel containers, this would result in a lapse in semantic interoperability between the reports of these two adjacent units. In this scenario, the sim-IS translation module should be designed to report fuel in two different ways, one for each unit, in order to simulate the semantic incompatibility of the information submitted by the units, due to their different processes.

For a pragmatic interoperability example, consider a scenario where CLB-1 is the training audience. CLB-1's processes are based on an expectation that all units' supply levels are updated in CLC2S every 6 hours. From the perspective of CLB-1's processes, one might argue pragmatic interoperability for sim-IS exchanges for unit supply levels requires all supply statuses in CLC2S to be a maximum of 6 hours old relative to the supply statuses of simulated units. In this scenario, CLB-1 is placed in direct support of a simulated rocket battalion and is responsible for providing timely ammunition resupply. The processes for the simulated high mobility artillery rocket system (HIMARS) battalion include reporting ammunition levels every other day. For this scenario, achieving sim-IS pragmatic interoperability requires supply statuses for the HIMARS battalion be submitted to CLC2S every other day, despite the resulting lapse in pragmatic interoperability between the rocket supply levels and the processes through which CLB-1 employs CLC2S.

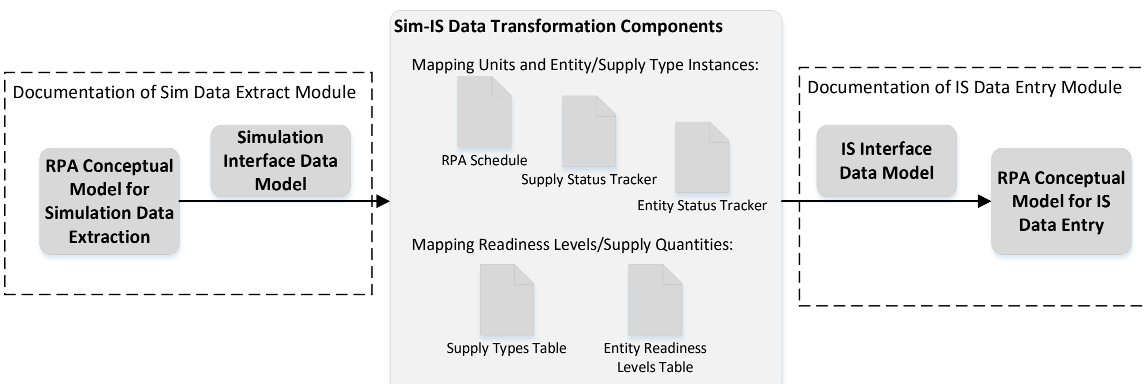
For another pragmatic interoperability example, consider a scenario where CLB-1 has received a request from CLB-5 to deliver fuel, using the CLC2S logistics support request function. This scenario applies to event-based sim-IS information exchanges. In this scenario, CLB-1's internal processes dictate that the unit delivering supplies executes a supply quantity transfer to the receiving unit in CLC2S upon completion of the mission. CLB-5's internal processes dictate that the receiving unit manually increments their supplies, and the sending unit manually decrements their supply levels following delivery of supplies. This combination of incompatible processes would result in an erroneous reporting of supply levels across the two units. To achieve pragmatic interoperability, the RPA workflows for the associated sim-IS environment should be designed to simulate the actions which would be taken by each unit relative to CLC2S, despite the resulting disparity between the ground truth supply levels tracked in the simulation and the supply levels manifesting in CLC2S.

6. RPA Information System Data Entry Modules

IS data entry modules enter translated and modified simulation data into a specific IS in a way that simulates data entry by an operational system user. This requires the identification of information entry requirements in terms of data syntax and semantics, but also identification of the process (or processes) commonly used by operational IS users. Data syntax and semantics requirements must be included in the documentation for an IS entry module so that the requisite data value mapping and formatting can be conducted by the data translation module. These considerations are addressed in the IS interface data model. The processes for navigating through the IS interface are captured in the RPA conceptual model for IS data entry. Additional considerations for the design of IS data entry modules include password management and the simulation of human IS user processes.

a. IS Interface Data Model

In this research, we use "interface data model" to describe the location and format of data extracted from simulations and requirements for data to entry into operational information systems. The mapping of a system's user interface data model to the central data transformation components is essential for ensuring the translated data is compatible with the IS fields into which it must be entered. The simulation and IS interface data models document the relationship between user interfaces and sim-IS data transformation components, informing RPA module design and maintenance. Figure 37 illustrates how simulation and IS interface data models relate to RPA conceptual models and sim-IS data transformation components in the mapping and translation of simulated entity and supply statuses across the sim-IS environment.



Simulated entity and supply instances and statuses are mapped across the sim-IS environment through RPA conceptual models, simulation & IS interface data models, and sim-IS data transformation components.

Figure 37. The Role of Simulation and IS Interface Data Models

The interface data model can be described in terms of keys and value entry fields. Keys identify the IS data elements necessary for applying status updates to a specific unit and entity instance or supply item type in the IS user interface. For the supply status updates in CLC2S, for example, keys include the unit name and the national stock number (NSN), which serve as unique identifiers for navigating to the correct field for entering a supply status update. The interface data model identifies how these keys manifest in the GUI, in terms of both location and data format.

Value entry fields identify the location and data format requirements for entry of status updates within a given IS. In CLC2S, supply status entry value fields require the entry of a non-negative integer. For CLC2S entity readiness status updates, keys include the unit name, equipment class, and equipment serial number, while the entry value fields consist of drop-down menus with specific options. Table 9 illustrates elements of the IS interface data model for CLC2S which are required for supporting the exchange of supply status information across a sim-IS environment. The complete interface data model for CLC2S is provided in Appendix B.

Table 9. Excerpt of CLC2S Interface Data Model for Supply Status Information Exchange.

IS Interface Data Model (CLC2S)			
IS Element	Location in IS GUI	Data Type & Formatting Requirements	Data Transformation Component Source
Entry Event: Initial Unit Selection for Supply, Equipment, or Personnel Status Updates			
Key: Unit	User Profile -Change Unit -New Unit	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	For Supply Entry: Supply Status Tracker: "Unit_Name" For Entity Status Entry: Entity Status Tracker: "Unit_Name"
Entry Event: Supply Status Entry			
Key: Unit	Assets -Manage Supplies -Search Supplies -"Unit"	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	Supply Status Tracker: "Unit_Name"
Key: Supply Type	Assets -Manage Supplies -Search Supplies -"NSN"	Free text	Supply Status Tracker: "Supply_Type_NSN"
Value Entry Field	Assets -Manage Supplies -Modify Unit Supply Item -Inventory Information	Free text: Non-negative integer, less than 100,000,000	Supply Status Tracker: "Modified Status"

The interface data model maps the key and entry value location and data formats to the values specified in the supply status tracker and entity status tracker files. The RPA conceptual model defines the processes to be executed by the IS data entry module, to simulate the actions of human operators of the target IS. This includes documentation of the specific process(es) for navigating through the IS GUI to the data entry locations, including identification of IS interface attributes and anchors which support the RPA's navigation through the IS GUI.

b. RPA Module Design for IS Data Entry: Simulating Human IS User Processes

The RPA module for IS data entry accomplished the final step of the sim-IS information exchange process: entering the data into the IS interface. It is designed to exchange the requisite data from the sim-IS data transformation components (supply status tracker and entity status tracker) in accordance with the IS data entry requirements documented in the IS interface data model. This RPA module is also designed to enter the requisite data into the IS interface in such a way that it simulates data entry process of the associated integrated business process, including human data entry.

In the literature, RPA workflows are often described as "simulating human processes." This is often an imprecise use of "simulate," as the objective of most RPA modules is not to simulate human-IS interactions, but to automate interactions through GUIs conventionally used by human users. This is an important distinction because, unlike conventional RPA workflows, the IS data entry modules for sim-IS environments are intended to simulate data entry as it occurs, or is expected to occur, in the associated integrated business processes. As RPA proliferates, there is a potential requirement for future sim-IS environments to simulate both human-IS interactions and operational RPA workflows. For the former, IS data entry modules must be designed to simulate the processes through which human IS users enter data. For the latter, IS data entry modules may actually employ real-world RPA modules, with some modifications to facilitate interactions with simulation data extract modules.

Designing an IS data entry module to simulate the process(es) conducted by human operators requires a model of IS-user processes for navigating the IS GUI to enter various types of data. Adequately simulating human processes does not necessarily require following the same exactly steps as human operators. In CLC2S, for example, supply statuses may be updated either through CLC2S GUI interactions or by loading an excel "feed" file. Human operators may use either process, though the feed files are less intuitive. Furthermore, both processes result in additions to the supply item's "Inventory History" reports which may be used by commanders and staff to review supply status changes. It is incumbent on the sim-IS environment designer to determine whether the process modeled by an IS entry module is adequate for their sim-IS environment.

Unfortunately, some operational ISs are fielded without standardized, universal processes for their implementation. In these situations, the designers of sim-IS environments are forced with a sort of "chicken or the egg" problem. The design of processes for leveraging ISs in context requires a sim-IS environment for the development and testing of processes, IS, and human competencies. Without an established integrated business process as the model, however, what should serve as the initial model for the design of the sim-IS environment? Considering the development, or coevolution, of cognitive systems is an iterative process, it may be best not to get hung up on the initial design of the sim-IS environment. Instead, in lieu of starting off with an ideal integrated business process, it may be best for sim-IS environment designers to push forward with creating a sim-IS environment which simulates processes and associated dynamics which are feasible for the existing organization. This will get the process started and provide a baseline for iterative development of the integrated business process.

For this research, no standard process exists for updating logistics statuses within CLC2S. Marines may update statuses through the "feed" files or by navigating through the CLC2S interface. Supply status transfers between units may documented through the "supply transfer" function, or separately by each unit's manual increment/decrement of supplies on hand. In lieu of a standardized process, the process which appeared the most intuitive to the researcher and the MCLOG M&S staff was selected for this research: updating supply and equipment statuses by navigating through the CLC2S GUI. This approach for updating supply and entity statuses requires no knowledge of CLC2S feed file structure and formatting requirements and requires no coordination between units regarding which unit is executing a supply transfer in CLC2S and when.

The IS data entry RPA module is developed independently from the simulation module(s) with which it may be coupled. The use of common sim-IS data transformation components facilitates reuse of IS data entry RPA modules with different simulations, though they do not guarantee semantic and pragmatic interoperability. These higher levels of interoperability must be evaluated independently for each sim-IS environment, including translation and modification module design considerations discussed previously.

These RPA modules may also be built using any of a variety of RPA platforms. While the specific capabilities of the RPA platform may vary, they may be expected to require some of the same details to interact with the target IS interface. GUI attributes and anchors must be identified which will facilitate navigation to the appropriate URLs, selection of the right buttons, and entry of data into the appropriate fields. The challenge of designing conceptual models of integrated business processes and IS GUIs and using such models to inform the design of RPA data modification and entry modules is discussed further in Chapter VII.

c. Password Management

When designing RPA workflows, a pair of linked decisions include the password management approach and whether the workflow will be attended or unattended. Unattended RPA workflows are executed without direct oversight either on local computers or geographically separated servers. These RPA workflows can be assigned their own passwords for various information systems with which they interact. Organizations which are unwilling to accept the risks associated with unattended RPA instances can conduct RPA workflows, which are executed with direct oversight of an individual or team.

For attended RPA, a process for maintaining oversight of RPA workflows should be developed and the IS data entry module should be designed to allow users the time to enter passwords for accessing requisite information systems. These IS entry modules may also be designed to mitigate system time-outs. A work-around can include interacting with a given IS at regular, frequent intervals to prevent the IS from logging out. For this research, an attended approach was implemented. The operator logs into the IS when the sim-IS environment is launched, entering their password after the RPA instance enters the username. The RPA module for data entry interacts with the CLC2S GUI at regular intervals by clicking on a GUI button to refresh the interface and prevent account logout for the duration of the sim-IS environment.

7. Simulation Data Extraction Modules

Like IS data entry modules, each RPA module for simulation data extraction is built for a specific simulation. Once an RPA module is built for a given simulation, it can be reused for other sim-IS environments, coupling the simulation with different information systems. Unlike IS data entry modules, simulation data extraction modules are not intended to simulate the interactions of pucksters with simulation interfaces. They are intended to support simulation of the target BP-IS process(es) as defined in the sim-IS environment conceptual model.

In some instances, the way a simulation is leveraged may influence the design of a simulation data extraction module, potentially necessitating redesign of the simulation data extraction module to ensure alignment with the sim-IS environment conceptual model. In a sim-IS environment which includes a JCATS, for example, fuel storage may be simulated in multiple ways. If unit bulk fuel stores are simulated through the use of a single entity which serves to simulate the unit supply node, then the simulation data extraction module must be designed to extract the supplies associated with that entity and save them as the associated unit's supply statuses. Another means of simulating bulk fuel stores is to allocate fuel to the different bulk fuel storage assets owned by the unit in accordance with their capacities (e.g., up to 5,000 gallons in the M970 fuel tanker and 850 gallons in the fuel SixCon). For this research the former approach was taken, as that is the approach currently taken by the MCLOG simulation team. If a sim-IS environment designer intends to use the latter approach, the JCATS data extraction RPA module would require modification. The challenge of aligning RPA modules with the sim-IS environment conceptual model is addressed further in Chapter VII.

For a simulation to be compatible with RPA-based sim-IS architecture, the simulation must have unique identifiers which are associated with entity and supply statuses in the simulation's user interface and which can be mapped to common unit/entity unique identifiers and supply type identifiers through the sim-IS data transformation components. This topic was addressed at the beginning of the chapter, but the identification of unique identifiers for a given simulation or IS is conducted during the development of their respective data extraction or data entry RPA modules.

An additional consideration for simulation data extraction module design is the different approaches for use of unique identifiers in entity and aggregate simulations. For entity level simulations, like JCATS, the simulation data extraction module must support extraction of each entity using its unique identifier in order for the translation RPA module to associate it with the equivalent entity status for the associated IS. For aggregate simulations, like MTWS, where individual entities are not simulated, entity and supply statuses are selected based on unique unit identifiers and the entity class and supply class identifiers. From there, the module uses the entity unique identifiers listed in the sim-IS data transformation components (supply status tracker and entity status tracker) to update entity statuses to mirror the distribution of entity statuses extracted from the aggregate simulation.

As with the development of IS data entry modules, the design of the simulation data extraction module depends on the development of a user interface data model for the simulation. This includes the identification of how the unique identifiers and entity/supply class indicators manifest in the simulation user interface and how they relate to the corresponding elements within the sim-IS data transformation components.

Lastly, the design of the simulation data extraction module must be conducted relative to the sim-IS StartEx data generation process. Identifying the location and formatting of unique identifiers and values in the simulation requires an understanding of the simulation's parametric data. Simulation parametric data defines attributes of supply and entity classes within the simulation. This includes entity speed, armor, and carrying capacities. It also includes the nomenclature and NSNs for the entity and supply classes, and how they manifest in the simulation user interface. While the StartEx data generation process is important for the development of any distributed simulation or sim-IS environment, this requirement for mapping the indicators for entity classes and supply types adds another level of complexity which is not required for conventional distributed simulation interoperability.

B. SIM-IS STARTEX DATA GENERATION AND THE BFG ARCHITECTURE

The generation of StartEx files for the simulations and associated information systems of a sim-IS environment is tedious, requiring the development of matching unit structures, unit and entity identifiers, and supply types and quantities. While the generation of such synchronized sim-IS StartEx files can be accomplished manually through conventional means, typing out feed files for a common scenario is time and manpower intensive with significant risk for human error. For this research, a process for automating the creation of synchronized simulation, information systems, and RPA files (RPA schedule, supply status tracker, entity status tracker, and total entity status tracker files) was developed, building upon an application previously built during the author's time serving as a modeling and simulation officer for MCLOG. This section identifies how the generation of StartEx data for RPA-based sim-IS environments differs from conventional StartEx data generation processes and provides an overview of the process.

1. A Brief Review: StartEx Data Generation for Simulations

Once a set of simulations have been selected for use in a distributed simulation environment, the designers of the environment must identify which units and entities will be controlled by which simulation and build the StartEx data accordingly. From there, the type of distributed simulation architecture employed for the environment determines the type of StartEx data synchronization required. For distributed simulation environments using only DIS for interoperability, the distributed simulation developers must ensure common DIS enumerations exist in the enumeration tables of the participating simulations for all entity types to be used in the scenario. DIS enumerations facilitate the identification of an entity's class. If entity enumerations are not synchronized prior to the start of an exercise, entities may be incorrectly interpreted by different simulations. In Figure 38, a JCATS F/A 18D aircraft is misinterpreted as an M1A1 by the Virtual Battlespace (VBS) simulation due to a misalignment in JCATS and VBS enumerations tables.

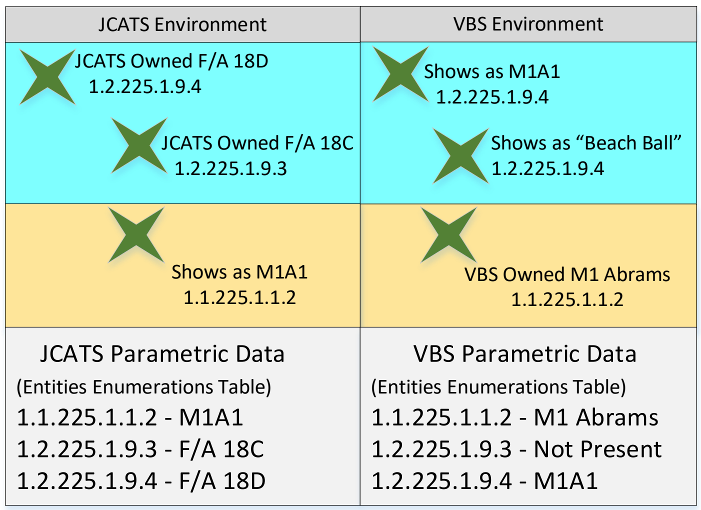


Figure 38. Aligning DIS Enumerations Across Distributed Simulations.

The Joint Live Virtual Constructive (JLVC) federation object model (FOM) is a high level architecture FOM used extensively throughout the DOD. This FOM extends the Real-time Platform Reference FOM, and the development of JLVC federations often encounters the same challenges of enumeration misalignment (and additional nomenclature or NSN misalignment)(Bowers et al., 2011). The obstacle of ensuring parametric data alignment for every JLVC federation instance was addressed within the DOD with the development of a simulation scenario and parametric data repository (the Joint Training Data Services [JTDS]) and standardized xml structure for populating simulations (Order of Battle Service [OBS]) (Bowers et al., 2011). This service supports the initialization of JLVC federations with synchronized unit structures, entity assignments, and supply allocations for each unit.

The JTDS site serves as a repository of entity and supply classes, terrain files, and scenarios. The OBS XML schema enables specification of a scenario (e.g., unit structure and allocation of simulated entities and supplies assigned to each unit) and initialization of that scenario data across JLVC federates. Unique identifiers are assigned to each unit and entity in the OBS (e.g., entity LVC ID) and the owning simulation for each unit and entity is specified. For federates to reflect the scenario specified in the OBS, the entity class enumerations and supply type information (e.g., NSN) in each federates parametric data must align with the enumerations and other supply/entity class information specified in the OBS.

One could be forgiven for expecting that the synchronization of parametric data and standardization of enumeration tables would be a simple, inconsequential affair. After all, the SISO Reference for Enumerations for Simulation Interoperability (SISO-REF-010-2021) specifies DIS enumeration value. Unfortunately, existing standards and tools are insufficient for ensuring common enumeration tables and parametric data alignment across DOD simulations. Simulation parametric data (and even JTDS) is not always kept up to date with the SISO standard. The SISO enumerations standard itself is also sometimes absent critical equipment, as the development of new equipment can outpace the assignment of new enumerations. This leaves local simulations managers to generate their own enumerations to make it through the next exercise (Bowers et al., 2011). These enumeration misalignments can result in entities specified in the OBS failing to manifest in one or multiple simulations. It can also result in entities manifesting in a federate as some entity class other than what is intended, like the F/A-18D to M1A1 example from Figure 38.

Although some simulations have reports for identifying entity enumerations which are not available in the parametric data (e.g., MTWS) these reports do not identify entity class misalignments. There is no automated way to determine whether the wrong enumeration has been assigned to an entity class in the simulation's parametric data, the JTDS database, or both. Ensuring alignment of parametric data between federates and the OBS file cannot be automated and instead requires tedious comparison of entity enumerations and other supply/entity class information (e.g., nomenclature, NSN, model) between the OBS (or originating software's database) and the parametric data of each federate.

2. StartEx Data Generation for Conventional Sim-IS Environments

For sim-IS environments, simulation StartEx data must be mirrored in the operational information systems to ensure the training audience is given an appropriate understanding of the battlefield. For map-based C2 systems which only show unit locations and names, this can often be handled by the simulation(s) with some middleware. For many information systems which must mirror the unit structure and entity/supply statuses (e.g., logistics, medical, or manpower information systems), however, the initialization of simulations and ISs with synchronized StartEx data is more challenging. This is due, in part, to the limitations of JTDS and OBS in supporting sim-IS environments. While JTDS supports the initialization of scenario data across multiple simulations, it was not designed to support synchronized sim-IS StartEx data generation. As seen by MCLOG as recently as their December, 2021 staff training exercise, manually synchronizing simulation and logistics ISs for continuously evolving sim-IS scenarios includes a significant risk of Sim-IS misalignment (H. Viramontes, personal communication, June 30, 2022).

Sim-IS StartEx data generation first requires identification of which entity and supply attributes are used in each simulation's parametric data for the selection of the entity or supply item when the OBS is loaded. This process, which we refer to as sim-IS StartEx data initialization, ensures the appropriate entities and supplies are allocated to the appropriate units at the beginning of the simulation. Sim-IS StartEx data initialization also requires consideration of how dynamic sim-IS information exchange will be conducted during the exercise, whether by RPA bots or human pucksters, by identifying how each entity or supply type will be represented in the simulation and IS GUIs. Unlike multi-simulation environments, where standardized DIS and HLA protocols support sim-to-sim information exchange, sim-IS information exchange generally must be conducted through the user interfaces. If a scenario includes two variants of 7-ton trucks, say MK23 and MK27 models, but both are represented as "TRUCK-MTVR" in the simulation interface, the simulation operator will not be able to distinguish the statuses of the separate variants for entry in the associated IS. Sim-IS StartEx data initialization must therefore also consider how representation of entity classes and supply types and unique identifiers in the user interfaces of both simulations and ISs, to ensure sim-IS information exchanges can be conducted appropriately through the GUIs.

a. Sim-IS StartEx Data Initialization

The importance of ensuring synchronization of sim-IS StartEx data cannot be overstated. If unit hierarchies, equipment, personnel, and supplies do not match across the simulations and ISs, down to individual names and serial numbers, the training audience may focus on fixing discrepancies and deciphering what constitutes exercise "ground truth" before they can even begin to exercise their staff processes. Worse yet, the training audience may interpret the disparity between simulation "ground truth" statuses and the erroneous information present in their operational ISs as evidence that their operational IS is faulty.

The conventional process for populating ISs like CLC2S and TCPT for sim-IS environments is to build a scenario in the simulation and then use a simulation-generated report to guide the manual creation of the unit hierarchy, and assignment of entities and supplies, in the operational ISs. For aggregate simulations (e.g., MTWS), where individual entities are not simulated, the unique identifiers for equipment and personnel in the ISs are immaterial, as there is no requirement for mapping them to the simulation entities. In these instances, only the alignment of entity class identifiers in StartEx data is necessary as dynamic synchronization of simulation and IS statuses requires total distributions of entity statuses per unit and entity class are aligned. For entity level simulations, however, the unique identifiers (e.g., names, electronic data interchange personal identifiers [EDIPI], serial numbers) for entities loaded in the IS must align with the unique identifiers for the mirrored unit and entity instances in the simulation(s).

Some entity-level simulations are not conducive to the prior specification of entity unique identifiers necessary for mapping of simulated entities to entity records in associated ISs. In the Joint Deployment Logistics Model (JDLM), personnel names and other unique identifiers are randomly selected by the JDLM simulation and cannot be defined in scenario StartEx files. This prevents the specification of personnel names and unique identifiers for entry in manpower ISs in advance of an exercise. JCATS presents a similar problem, as the only unique identifier, the JCATS ID, generated by JCATS when the scenario is loaded.

Both of these simulations require workarounds for assigning unique identifiers which can be used to map simulation entities (and their statuses) to entities in operational ISs. For JDLM, this can be done by changing a JDLM file which specifies personnel specialties to include individual personnel names and EDIPI. In this way, unique identifiers can be assigned to JDLM personnel which will align with personnel names and EDIPI in the operational IS. For JCATS, the entity role name, which is specified in the OBS, can be used as the unique identifier for both personnel and equipment. Unique identifiers are not manually generated by JTDS for the role name field, which makes the assignment of unique identifiers to role names within the JTDS interface a manpower intensive and error prone process.

b. Sim-IS StartEx Data Generation to Support Dynamic Sim-IS Synchronization

The conventional process for updating ISs with simulation statuses throughout a simulation, what we refer to as dynamic sim-IS synchronization, depends on the unique identifiers for entities and the mapping of both entities and supply types between the simulation(s) and IS(s). Unlike simulation-simulation interoperability, where dynamic exchange of entity statuses is automated through standardized protocols (e.g., DIS, HLA), dynamic sim-IS synchronization often includes a manual exchange of information by pucksters through both simulation and IS GUIs. The puckster must be able to associate an entity or unit supply status found in the simulation with the equivalent entity and unit supply status location in the operational IS. Sim-IS environment designers must identify the process for pucksters and provide aids for the translation of entity readiness levels and supply status quantities between the simulation(s) and IS(s), including semantic differences (e.g., cases versus individual MREs, bulk fuel stores versus all unit fuel).

The design of these manual sim-IS information exchange processes requires an understanding of how entity and supply information manifests in the simulation interface and the associated IS interface. This requires an additional step for StartEx data generation for sim-IS environments: ensuring the parametric data for simulations (and corresponding entity and supply class data in ISs) are aligned or explicitly mapped to one another. This additional step requires sim-IS environment developers to ensure entity/supply class indicators are aligned across simulation and IS parametric data for how these indicators will manifest in their respective user interfaces. For a JCATS-CLC2S environment, for example, this includes updating the supply class nomenclature within JCATS' Vista application to match the supply nomenclature specified in the CLC2S database, to ensure pucksters can update supply statuses appropriately.

During the execution of a manual sim-IS environment, pucksters require a means of mapping simulation entities and supply types and instances. Ideally, the representation of entity classes and supply types would be the same in each simulation and IS of a sim-IS environment. Consider a simulation GUI which identifies equipment entity classes by their model, showing a "MTVR - MK23," for example, where "MK23" is the model. It would be ideal for the associated IS to also identify this entity class by "MTVR - MK23," at least for the puckster tasked with manual sim-IS exchanges. If the IS GUI instead represents the entity class by table of authorized material control number (TAMCN) (e.g., D0198 for MK23), then the puckster may require a table which identifies how all simulation models align to IS TAMCNs.

This issue can be further complicated by differences in how supplies are tracked across simulations and ISs, which can be due to semantic differences or different units of issue. Chapter II introduced the challenge associated with representing unit fuel statuses across simulations and logistics ISs, which is an example of the former. During preparations for the Marine Corps' Large Scale Exercise 2017 (LSE), sim-IS environment developers identified this as a problem for achieving sim-IS exchanges between MTWS and CLC2S. Their solution was the provision of a spreadsheet to all pucksters which supported calculating the bulk fuel status for each unit by subtracting the cumulative capacity of all vehicle fuel tanks associated with a given unit from the total fuel associated with the unit. An example of different supply units of issue can also be seen in considering a MTWS-CLC2S environment. While rations are tracked within MTWS by the individual ration, they are often tracked within CLC2S by cases of MREs (or boxes of tray rations). Accounting for these differences in supply units of issue would similarly require a table be provided to pucksters for manually calculating the appropriate value to enter into the associated IS (e.g., translating 36 rations in MTWS to 3 cases of MREs for entry in CLC2S).

c. Notable Obstacles to Sim-IS Synchronization

Some common simulation practices do not lend themselves to clear sim-IS synchronization. These practices, which include the simulation of complex assets, fuel allocation, and supply part requisitions, can limit the ability of simulations to support simulation of integrated business processes. In some simulations, “complex assets” are used to represent the combination of multiple end items. This is so common that DIS enumerations for many vehicles include an option for specifying the weapon system associated with the vehicle (e.g., 1.1.225.6.1.37.2 is a M1116 with a M2, .50 caliber machine gun). While this is an efficient way of simulating a high mobility multipurpose wheeled vehicle (HMMWV) with a mounted machine gun, it becomes a problem when seeking to reflect simulated equipment statuses in operational ISs.

In the real world, weapon systems and communications equipment are often accounted for as distinct end items, regardless of the platform upon which they are mounted. Furthermore, damage to a vehicle does not necessarily mean the mounted weapon system or radio received the same damage. However, when simulations track all of the equipment as a single “complex asset,” it complicates the generation of synchronized StartEx data and the ability of pucksters to conduct status updates of the complex assets across a sim-IS environment. There is often no separate unique identifier for the machine gun or radio and no separate process for representing their readiness levels or simulating their repair.

A similar phenomenon is seen with the simulation of unit fuel supplies in aggregate simulations. As discussed earlier, it is difficult to determine how to report only bulk fuel statuses when a simulation tracks only a single fuel level which includes both bulk fuel stores and the fuel already allocated to vehicle tanks. For the Marine Corps' LSE 2016, pucksters were given tables which identified the fuel capacities for all vehicle types being simulated (Hensien, 2016). Before a puckster was to report the unit's fuel status, they needed to first calculate the total fuel capacity of all individual vehicle tanks which was then subtracted from the total amount of fuel for a unit. While this approach may be better than the alternative, it has its own shortcomings, as it assumes all vehicles are always topped-off with fuel.

Additional issues must be considered when designing sim-IS information exchange for the simulation of integrated maintenance and supply processes. Several simulations are touted for their ability to simulate maintenance and supply considerations. JDLM, for example, simulates the evacuation of equipment through maintenance echelons, the creation of supply requisitions for repair parts based on the maintenance defect, and the routing of said repair parts to the appropriate maintenance facility. While this initially seems ideal for exercising maintenance and supply processes, an issue arises when seeking to associate the maintenance work order and supply part requisition identification numbers.

In the Marine Corps, supply requisitions are associated with specific maintenance work orders. If a repair part arriving at a unit is needed by multiple vehicles, it must be hung on the piece of equipment for which it was ordered. The comparison of maintenance work orders and the tracking and prioritization of supply requisitions is a routine process known as maintenance and supply reconciliation. In JDLM, however, supply part requisitions are not associated with a specific work order number, only with the requesting unit. This prevents the simulation designer from being able to populate maintenance and supply information systems with the requisition information. This misalignment of JDLM with Marine Corps maintenance and supply processes was only realized by the researchers following attempts to generate integrated maintenance and supply reports. Ideally, the comparison of simulation and integrated business process conceptual models would support identification of this limitation earlier in the design of a sim-IS environment. This is discussed further in Chapter VII.

An additional problem arises when there are different levels of granularity in the representation of entity classes. For example, Marine Corps equipment is often tracked by either TAMCN or model number. These entity class identifiers do not always align one for one, however. The D0198 TAMCN, for example, refers to both MK23 and MK25 models of medium tactical vehicle replacement (MTVR) cargo trucks. The only difference between these vehicles is that the MK25 MTVR includes a winch. If a simulation represents the MK23 and MK25 as separate entities, while the associated IS tracks vehicles by TAMCN, the sim-IS designer must determine how to consolidate the reporting of MK23 and MK25 entities. This problem can be even more difficult if the issue is reversed.

These are examples of sim-IS design problems which should be identified during the conceptual analysis phase of sim-IS design and development. Some of these sim-IS alignment issues can be adjusted with workarounds (e.g., adding separate weapon and communication systems to complex assets for asset readiness reporting) and specified scenario simplifications (e.g., individual vehicle fuel tanks are assumed to be 80 percent fuel on average). Others present insurmountable obstacles for simulating integrated business processes without risking negative training (e.g., stimulating maintenance and supply systems with a simulation which does not appropriately simulate the integrated business process). While these obstacles to sim-IS environments must be addressed in conceptual analysis phase of sim-IS environment design and development, solving them is outside the scope of this research.

3. StartEx Data Generation for RPA-based Sim-IS Environments

StartEx data generation for any sim-IS environment requires the mapping of simulation and IS data models for initialization and dynamic synchronization of units, entities, and supplies. RPA-based sim-IS environments require two additional actions for StartEx data generation: specification of desired STS dynamics and creation of the RPA data transformation files. The former informs the design of the RPA modification module, and the latency and timeliness distribution files, for the simulation of data degradation across the sim-IS environment. The latter builds on the previously described sim-IS StartEx data generation requirements, as the RPA data transformation files provide an explicit mapping of simulation and IS parametric data for sim-IS exchange.

The specification of desired STS dynamics can take any form, so long as the description of the nature and frequency of desired data degradation captures all considerations necessary to inform the design of the RPA modification module to simulate the dynamic(s). The format of the RPA latency and timeliness distribution files depends on the design of the modification module, though, for the prototypes developed in this research, specification of latency and timeliness distributions simply requires latency and timeliness quantities be listed in separate excel files which are accessible by the RPA workflow. The number of times each value is specified determines its probability of occurrence, as the RPA modification module samples from these lists using a pseudo random number generator during sim-IS environment execution. The specification of STS dynamics for sim-IS environments is addressed further in the discussion of sim-IS conceptual modeling in Chapter VII.

RPA data transformation files can be thought of as roughly equivalent to the tables which inform puckster sim-IS exchanges. While the creation of these files requires some manual actions, like identifying the way entity class/supply type indicators manifest in GUIs, their generation can be largely automated once the structure of simulation and IS feed files are identified. For this research, the generation of the RPA data transformation files was automated through the extension of an application previously built by the author for the generation of synchronized sim-IS StartEx files: the Batch File Generator.

4. The Batch File Generator Application

Between 2015 and 2017, the MCLOG Modeling and Simulations department improved the inefficient, manual sim-IS environment StartEx data generation process by building a Java application which automates the populating of the four primary MLS2 and simulation systems then used for MCLOG BSTXs. This application, named the Batch File Generator (BFG), automates the process by translating unit information, as defined in a standardized excel template, into the requisite input file formats for each of the separate simulations and IS. These include eight different excel “feed files” for CLC2S, an excel

TCPT unit hierarchy file and another TCPT file for every unit in the hierarchy, three JDLM files, and a .txt "batch file" script for MTWS.

All (15) different types of files produced by the BFG are immediately ready for loading into the systems, providing a fully synchronized set of StartEx data for an exercise. Rather than the typical week or longer required for manual generation, the BFG generates these files in a matter of minutes and without the errors that inevitably come from the manual transcribing and translating data in such quantities.

The BFG application has been employed by MCLOG in support of many battalion, regiment, and Marine Logistics Group level staff training exercises. The automation of the sim-IS environment StartEx data generation process resulted in higher detail exercises and facilitated increased rehearsal time for MCLOG role players prior to execution of exercises. As the number of exercises conducted at distributed locations increased, the BFG proved even more crucial in facilitating fine tuning and tailoring of exercises for training audience requirements.

This automation not only cut exercise development time from weeks to minutes, it also facilitated more precise exercise design and enabled a previously infeasible degree of realism with the increased integration of logistics information systems. Without the BFG it would be too time and manpower intensive to load matching, unique equipment serial numbers and notional personnel information into CLC2S and TCPT. The design of the BFG allows these details to be generated in minutes, facilitating higher fidelity training for logisticians and administrative personnel than would be otherwise available with their real-world ISs.

5. Extending the BFG for OBS XML and RPA Data Transformation Components

In 2021, in support of this research and the MCLOG M&S team, the BFG was modified to accept an OBS XML file as an input. This was conducted to facilitate the application of the BFG in support of sim-IS StartEx data generation for entity-level simulations, and JCATS in particular, as the MCLOG M&S team began exploration of the JCATS simulation for their staff training exercises. While the previous BFG model included the generation of MTWS Batch Files and JDLM feed files for populating the simulations, the new version instead modified the OBS XML file itself. Modification to the OBS includes addition of unique identifiers for every entity (names and EDIPI for personnel, serial numbers for equipment) and can also include modification of additional attributes for personnel including gender, blood type, and religion. These unique identifiers and entity attributes are mirrored in the StartEx files, known as "feed files," for the CLC2S and TCPT logistics ISs. Figure 39 provides a high-level conceptual model of the updated BFG.

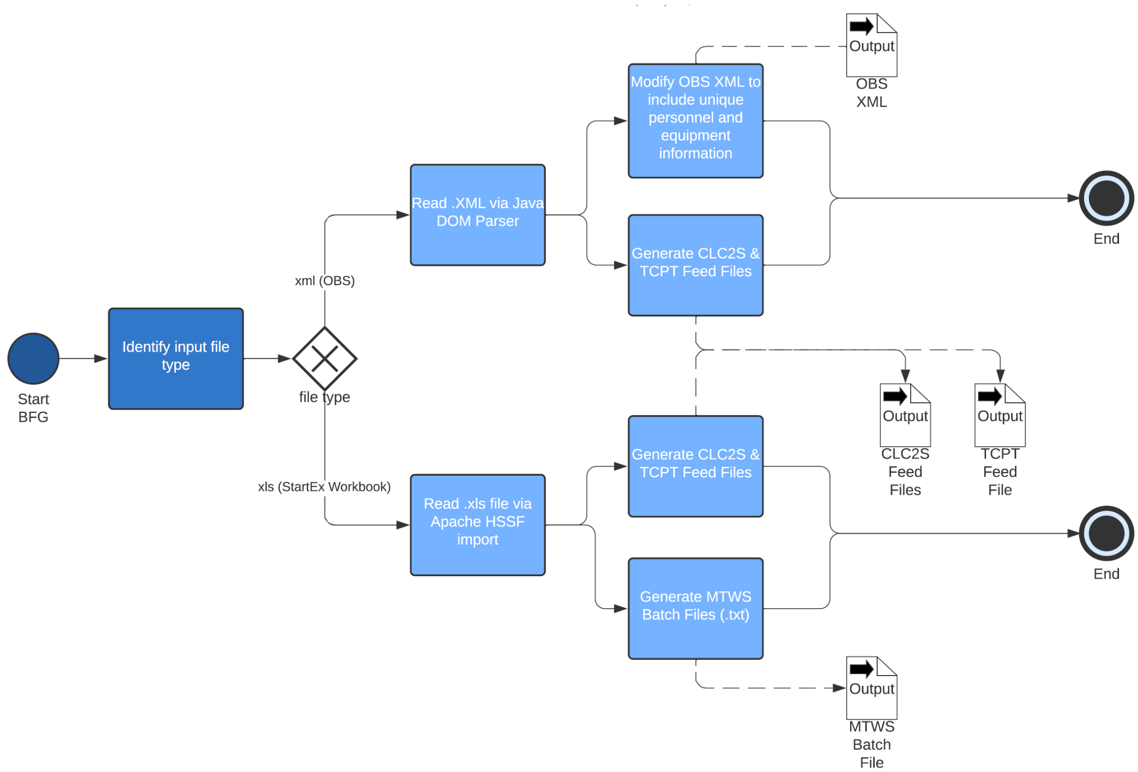


Figure 39. Batch File Generator Conceptual Model.

Using the OBS XML as the new input file for the BFG extends the utility for the BFG application beyond the MTWS and JDLM simulations for which it was originally designed. JTDS is managed by the Joint Staff J7 to support exercise development across the DOD, and the OBS XML files it generates can be used in numerous simulations. That said, the BFG is not a production quality application, and its use in support of additional simulations would require additional analysis and potential modifications. The BFG's modifications to the OBS for addition of entity unique identifiers was also designed specifically for JCATS, and it may require adjustment for use with other entity-level simulations. For example, while the OBS includes a field for equipment entity serial numbers, JCATS does not use this field, as there is no "serial number" field in JCATS, just a JCATS ID. A workaround for facilitating visualization of a serial number for each equipment entity in the JCATS GUI was achieved by having the BFG add a unique serial number to the role name field for each equipment entity in the OBS. If another simulation does use the "serial number" field in the OBS, the BFG should be modified to instead add the unique serial number to this field in the OBS.

In addition to generating synchronize simulation and IS StartEx files, the updated BFG was also updated to generate most of the sim-IS data transformation component files, including the RPA schedule, entity status tracker, supply status tracker, and total entity status tracker. The BFG does not populate the "Sim Supply Item Nomenclature" column in the supply status tracker, which identifies how the supply item type is identified in the simulation GUI. This must be manually determined by the sim-IS environment developer in relation to the simulation's parametric data. This task would be aided by ensuring documentation of the simulation data extraction module includes identification of how parametric data impacts the way unit, entity, and supply unique identifiers and attributes manifest in the simulation GUI. The BFG also leaves the "Sim ID Number" field blank in the entity status tracker. This field is used to associate the JCATS ID with the LVC IDs and serial number/EDIPI of each entity, and must be populated by the RPA middleware once the JCATS simulation has commenced.

Aside from the OBS, the only other file which must be adjusted for executing the BFG is a table which specifies TAMCN identifiers to be associated with each equipment entity class, by entity class name. For ISs which use TAMCNs, this ensures the appropriate TAMCN is specified for each entity class to be used in an exercise. The early creation of this table can also be of value to exercise designers, as the selection of multiple variants of a type of entity could confuse the exercise design and reporting. Consider a situation where multiple entity classes exist within JTDS for a M777 howitzer. If a designer accidentally assigns two different variants within the same unit, say 6 of one variant to Company A and 6 of another variant to Company B, with both tracked as E0671s (the TAMCN for M777s), errors in IS initialization and sim-IS exchanges can result due to different nomenclatures or other such simulation identifiers.

A similar problem can arise in the selection of supplies like ammunition for use in a scenario. If entities to be used within a scenario use different variants of the same class of ammunition (e.g., different simulation supply classes for linked .50 caliber rounds), an artificial ammunition supply and compatibility may arise, where a unit possesses the right ammunition for their weapon system, but a simulation supply class that is incompatible for the simulation entity class. A process for selecting entities and supplies within JTDS relative to the parametric data for the associated simulation is illustrated in Appendix D. These parametric data management considerations are not unique to RPA-based sim-IS environments. However, the development of sim-IS environments can be expected to require more thorough, efficient processes for ensuring proper parametric data alignment as these environments increase in size and complexity.

C. RPA-BASED SIM-IS ENVIRONMENT EXECUTION

The execution of RPA-based sim-IS environments is much less complicated than the design and development of the environment and execution of the sim-IS StartEx synchronization process. RPA-based sim-IS environments require minimal user oversight and provide the means for exercise control personnel to stay abreast of all scheduled sim-IS exchanges and pending/imposed data modifications in accordance with desired sociotechnical system dynamics. This architecture is also designed to handle various events which may occur that disrupt the workflows and require the RPA software to be restarted.

During a staff training exercise, the exercise control personnel and exercise observer/trainer/controllers (OTCs) must be aware of when the training audience is receiving information and whether that information has been modified. This is necessary so the exercise staff can identify how long it takes for staff to notice that the logistics statuses have been submitted and to identify whether the training audience identifies any discrepancies imposed. The RPA-based sim-IS environment architecture supports such exercise control oversight through the same files which guide the execution of the RPA workflows themselves.

By positioning the RPA schedule, supply statuses tracker, and entity statuses tracker files in a shared file location, the exercise control personnel can access these reports at any time. The RPA schedule, which shows the times for all supply and entity extract events (from the simulation) and entry events (into the IS) only needs to be pulled at the beginning of the day, assuming daily status updates. With the RPA schedule being essentially static throughout a given day, exercise personnel will know when supply and entity status updates will be submitted. They also know when statuses will be extracted, translated, and modified, and they can check the entity status tracker and supply status tracker files accordingly to see if any data modifications were imposed in accordance with the specified error types and probabilities of occurrence.

During execution, RPA-based sim-IS exchanges may be disrupted by network outages, server accessibility errors, or even user error (e.g., when a person tries to use the computer hosting an RPA instance being executed). The RPA-based sim-IS environment architecture presented here is designed to be resilient against these types of disruptions. The RPA Schedule events and entity/supply status values are continuously saved to the data transformation files throughout the execution of a sim-IS environment. This allows a user to restart the sim-IS environment following a disruption event, without concern that this information has been lost. Upon restarting the RPA workflow, the RPA Schedule will pick up where it left off with the exchange of sim-IS data.

It is possible that RPA modules may be written which can handle any exception such that a shutdown and restart of the RPA middleware is unnecessary. The prototypes developed for this research are not production quality, however, and the architecture was designed to support unhandled exceptions.

V. RPA-BASED SIM-IS ARCHITECTURE VERIFICATION

A mixed methods approach is used for the verification and validation of the proposed RPA-based sim-IS architecture as a new means of sim-IS information exchange and simulation of STS dynamics. Verification is defined as "the process of determining that a model or simulation implementation and its associated data accurately represent the developer's conceptual description and specifications" (USD(AT&L), 2009, p. 10). Verification of the RPA-based sim-IS information exchange approach is conducted through simulation of sociotechnical system dynamics specified in coordination with the research sponsor. These simulations were conducted in a controlled environment, the MOVES training and simulations lab, and simulated sociotechnical system dynamics in four experiments across two sim-IS environments: JCATS-CLC2S and MTWS-CLC2S. The sociotechnical system dynamics simulated were defined in coordination with the MCLOG M&S team.

This verification of the architecture for the specified STS dynamics is necessary but not sufficient for validation of the architecture supporting the research sponsor's requirements. Validation of the approach is achieved by supplementing the quantitative verification of sim-IS information exchange and STS dynamic simulation with qualitative, subject matter expert validation of the architecture in the form of a demonstration and interviews with the research sponsor. Field demonstrations and interviews were conducted with training organizations aboard Marine Corps Air Ground Task Force Training Center, Twentynine Palms, for qualitative validation of the architecture relative to sponsor requirements for employment in support of training environments. Interviews conducted with simulation professionals of MCLOG and the MCAGCC Battle Simulation Center supported qualitative validation of the architecture as a viable means of simulating the requisite sociotechnical system dynamics in simulation supported staff training environments. Validation of the RPA-based sim-IS architecture through qualitative methods is discussed in Chapter VI.

The simulations (MTWS and JCATS) and logistics IS (CLC2S) used in the two sim-IS environments were selected based on their relevance for the sponsor. MCLOG has historically relied on the MTWS constructive simulation to support its staff training environments. The MCLOG M&S team has been experimenting with using the constructive simulation JCATS in recent years as a potential replacement for MTWS in support of the staff training exercises they conduct. A key difference between these simulations is that MTWS is an aggregate simulation while JCATS is an entity-level simulation. The use of these simulations presents an opportunity for this research to identify unique opportunities and challenges associated with employing an RPA-based sim-IS information exchange approach with aggregate and entity-level constructive simulations. The Marine Corps' primary information system for tactical logistics C2, CLC2S, was selected as the IS for both sim-IS environments.

A. SIM-IS ENVIRONMENT CONCEPTUAL MODEL

This research uses the process for submission of logistics statuses (LogStat) reports by notional Marine Corps units to validate the proposed RPA-based sim-IS architecture. A reference model for the LogStat submission process is presented in Figure 40. This is a high-level abstraction of the process. The specific types of errors and the effects of timeliness variability on the LogStat entry into CLC2S are specified in the next section. A simplification of the LogStat submission process (Figure 41) is used for the sake of simplicity in the design of the sim-IS environments used in this research (Figure 42), including four specific simplifications discussed below. Simplifications include:

1. All supply and entity status measurements are simulated as occurring at the same time per unit.
2. All supply and entity status entries into the appropriate logistics IS are simulated as occurring at the same time per unit.
3. LogStat submission is simulated as occurring once daily per unit, with a common target submission deadline of 17:00 for all units.
4. For MTWS-CLC2S environments, all supplies allocated to simulated units are assumed to reflect unit bulk supplies only.

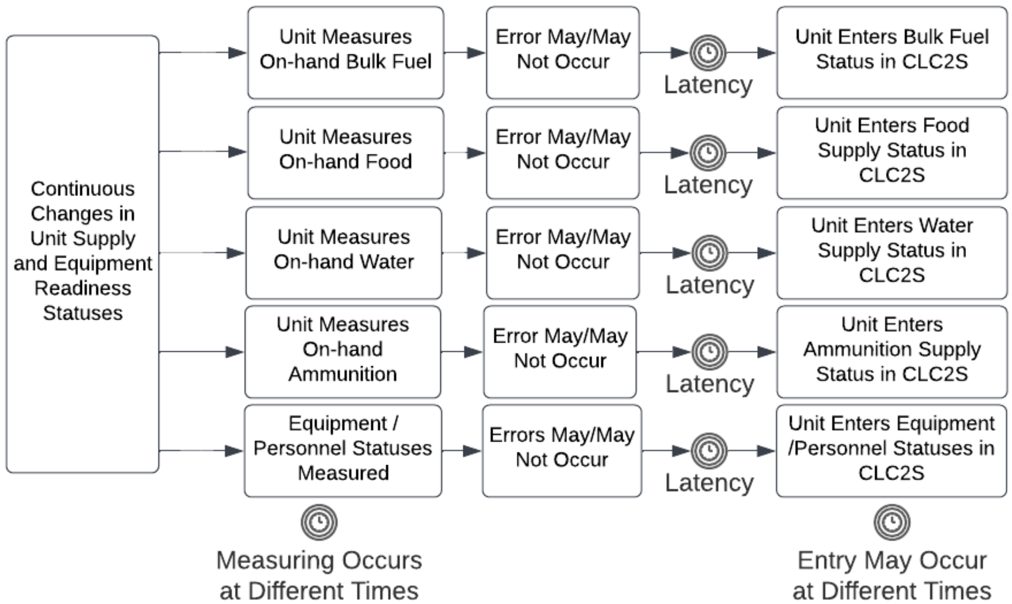


Figure 40. LogStat Submission Process Reference Model.

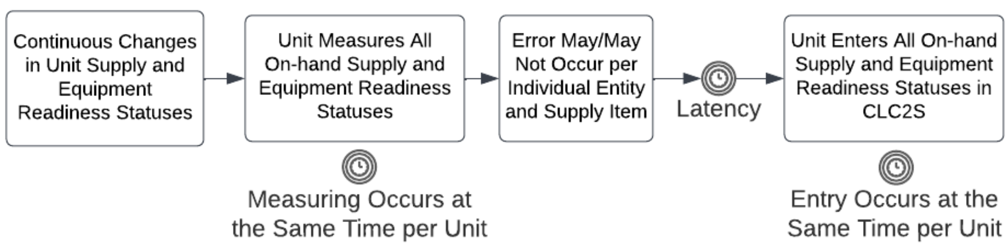


Figure 41. LogStat Submission Process Conceptual Model.

Two of the simplifications relate to the dynamics of the integrated business process itself, as seen in the transition from the reference model (Figure 40) to the conceptual model (Figure 41). First, the measurement of each unit's on-hand supply and equipment/personnel readiness statuses are assumed to occur at the same times. This results in extraction of unit supply levels and entity readiness statuses from the simulation (MTWS or JCATS) being scheduled to occur at the same times per unit. This assumption simplifies the structure of the RPA schedule as well as the internal actions required of the RPA workflow in tracking when the statuses of each individual supply item and entity must be extracted. Second, each entry of each unit's supply and equipment/personnel readiness statuses in CLC2S are assumed to occur at the same time. This results in entry of unit supply levels and entity readiness statuses into the IS (CLC2S) being scheduled to occur at the same time per unit, similarly simplifying the RPA schedule and workflow. LogStat submission is also assumed to occur once daily, with the times for entry of LogStat occurring with some degree of timeliness variability around a common LogStat submission deadline (e.g., 17:00).

Figure 42 shows how the different components of the LogStat submission process conceptual model (Figure 41) are distributed across a simulation (MTWS or JCATS), RPA workflow, and IS (CLC2S) in the sim-IS environments used in this research. The fourth simplification relates to how one of the simulations (MTWS) supports the simulation of the LogStat submission process in a sim-IS environment. JCATS supports identification of unit supply levels that are distinct from the supplies that have been distributed to individual personnel and equipment (e.g., bulk fuel versus fuel in individual vehicle tanks), by allocating all unit bulk fuel to a designated entity. MTWS does not support such distinct reporting of units' supply levels. Instead, MTWS provides the simulation user with a single value which encompasses both bulk supplies and the supplies distributed to personnel and equipment across the unit. For purposes of this research, all supplies represented within units in MTWS are assumed to represent the unit's supply stores, with no distribution of fuel or other supplies to personnel or equipment.

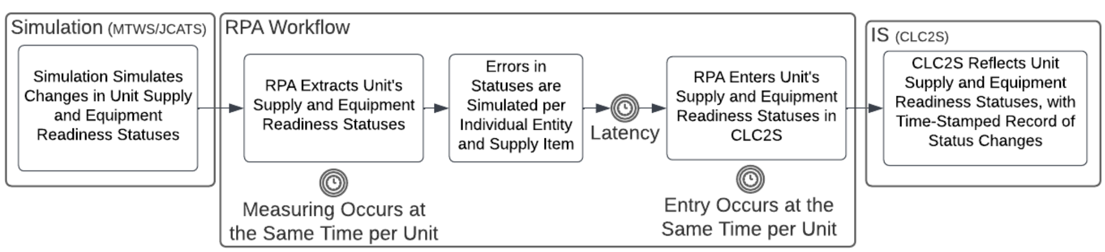


Figure 42. Conceptual Model of Sim-IS Environment for Simulation of LogStat Submission Process.

These simplifications represent one way a sim-IS environment may be designed to simulate the target LogStat submission process. Such simplifications are common and necessary for the design of sim-IS environments. The first and second simplifications, for example, are often seen in sim-IS information exchanges executed via pucksters, as they can be expected to make sim-IS information exchanges simpler and less time intensive. While the simplifications made for the sim-IS environments used in this research represent but one way sim-IS environments may be designed for representation of the target environment and associated STS dynamics, they do not constrain the generalizability of the results to different sim-IS environment designs. The specific RPA modules presented in this research are not intended to be a panacea for all sim-IS environment designs in their existing form. Instead, they demonstrate how RPA modules can function in the proposed RPA-based sim-IS architecture to support the specified intended use. These modules support verification and validation of the RPA-based sim-IS architecture through the simulation of the specified dynamics for the given sim-IS environment. Taking different approaches for the simplification issues addressed here would require some minor changes to different RPA modules, which could be maintained in a library of RPA modules for different simulations to support the different ways the simulations may be employed by different organizations or for different sim-IS environments. By maintaining conceptual models for each RPA module in such libraries, sim-IS environment designers would have the flexibility to reuse RPA modules when developing sim-IS environments in accordance with a target sim-IS environment conceptual model.

B. VERIFICATION OF RPA-BASED SIM-IS ENVIRONMENT ARCHITECTURE

This section identifies how target sim-IS information exchange dynamics, specified for simulation in a lab environment, are used to support verification of the RPA-based sim-IS architecture relative to hypotheses 1, 2, and 3. The target sim-IS information exchange dynamics are specified, as are the measures used for verifying the simulation of each dynamic and the thresholds for testing the associated hypotheses. Lastly, the structure of the controlled simulation of the RPA-based sim-IS architecture is described along with a description of the data collection processes.

1. Specifying Target Sociotechnical System Dynamics for Simulation

The sociotechnical system dynamics used for verification of the RPA middleware were defined in coordination with the research sponsor, MCLOG, to approximate some of the dynamics which logistics staffs may encounter when supporting military operations. The simulated information exchange dynamics consist of temporal dynamics and information content quality degradation dynamics. Temporal dynamics include information latency (Figures 43, 45, and 47) and timeliness variability in the submission of LogStat reports (Figures 44, 46, and 48). Latency represents the time between obtaining a unit's supply status (e.g., physically counting pallets of food, consolidating ammunition status reports from disparate subordinate units) and entering that information into the requisite IS. Timeliness represents the time between the deadline for LogStat submission and the time when a unit actually submits its LogStat report (e.g., submitting information two hours early or five hours late due to communications issues or operational limitations).

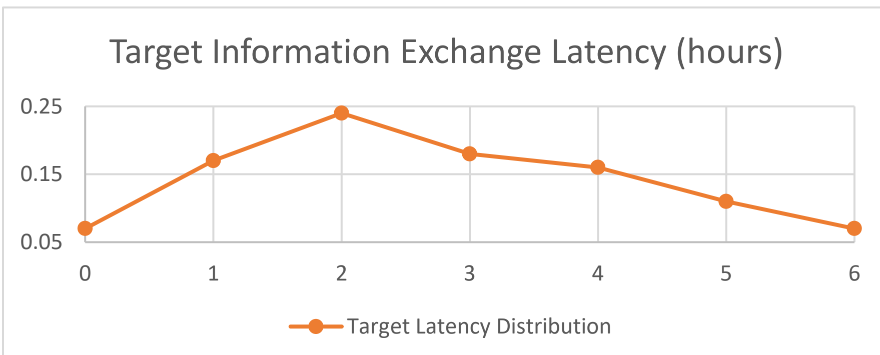


Figure 43. Target Latency Distribution for JCATS-CLC2S Simulation and MTWS-CLC2S Simulation 1 (Unimodal).

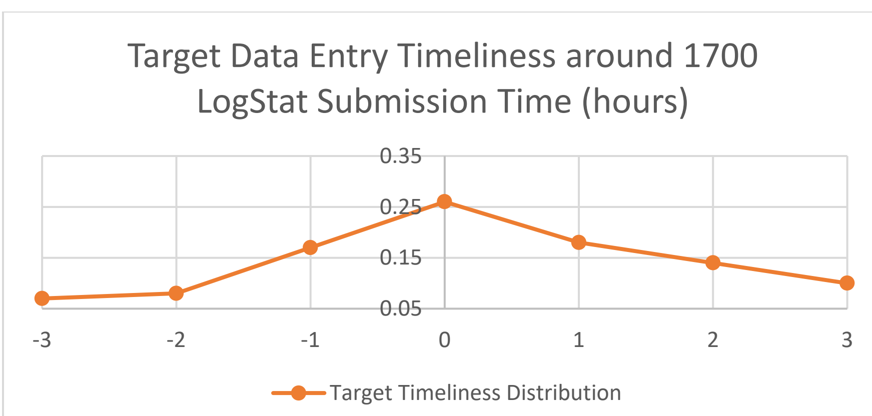


Figure 44. Target Timeliness Distribution for JCATS-CLC2S Simulation and MTWS-CLC2S Simulation 1 (Unimodal).

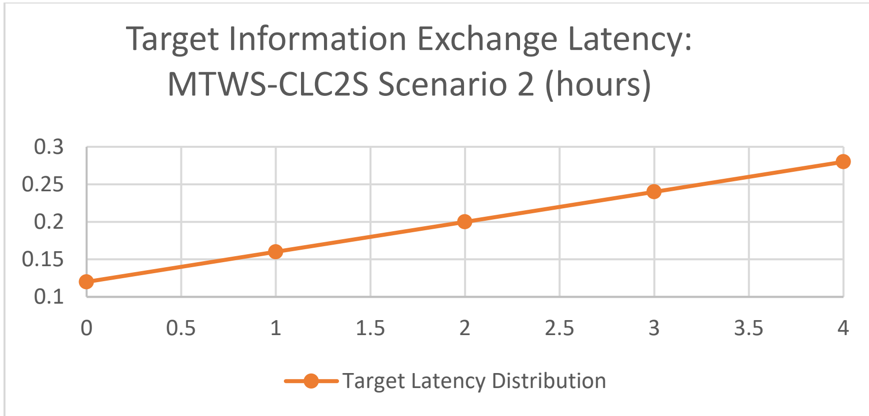


Figure 45. Target Latency Distribution for MTWS-CLC2S Simulation 2 (Linear Distributions).

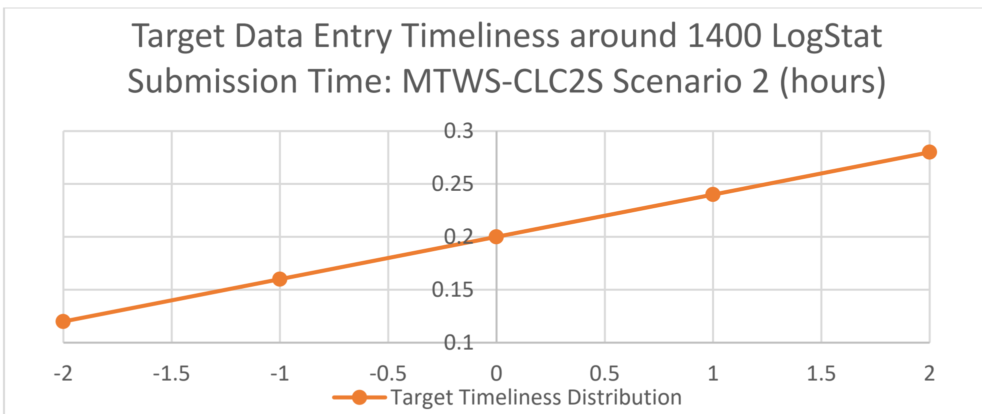


Figure 46. Target Timeliness Distribution for MTWS-CLC2S Simulation 2 (Linear Distributions).

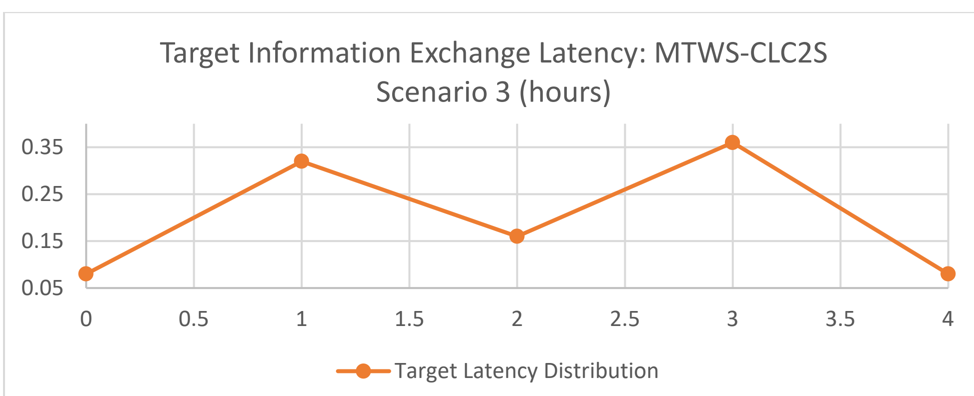


Figure 47. Target Latency Distribution for MTWS-CLC2S Simulation 3 (Bimodal Distributions).

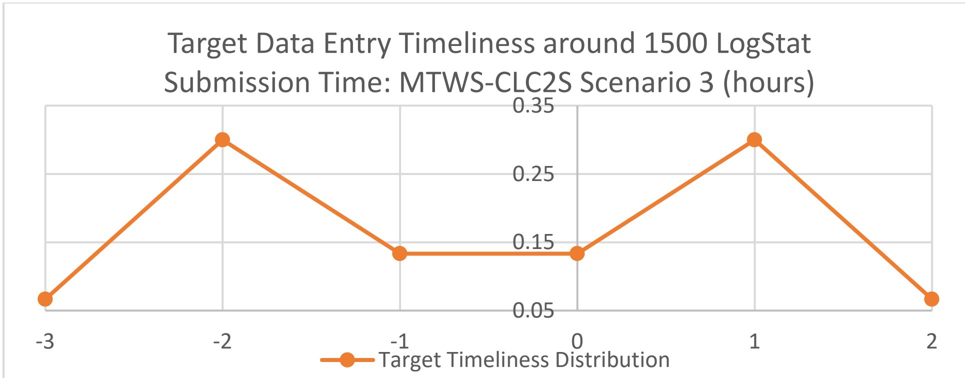


Figure 48. Timeliness Distribution for MTWS-CLC2S Simulation 3 (Bimodal Distributions).

The first latency and timeliness distributions, shown in Figures 43 and 44, were developed in coordination with the research sponsor, MCLOG, to approximate the levels of latency and timeliness variability anticipated in daily logistics status report submission. Additional temporal distributions were generated to test the tunability of the RPA workflows for simulation of different temporal dynamics as required by stakeholders. These include linear distributions of latency for timeliness (Figures 45 and 46) and bimodal distributions for latency and timeliness (Figures 47 and 48). Frequency of submissions in all simulations is limited to once per day, though the LogStat submission time around which timeliness variability is applied is varied between simulations to test tunability of submission times. Less frequent reporting, which may be of benefit in supporting future analysis, as discussed in Chapter II, would result in insufficient opportunities for exercising staff processes, as staff training exercises often last only three to five days.

Information content quality degradation is simulated in terms of data accuracy, precision, and completeness, with probabilities of occurrence specified in Table 10. Degraded information accuracy is simulated through the entry of erroneous data in the IS by adding an extra digit (zero) to the number extracted from the simulation. Degraded information precision is simulated by rounding values to the nearest hundred for quantities greater than five hundred. Degraded information completeness is simulated by intentionally skipping the entry of updates for supply statuses. Verification of the RPA-based sim-IS architecture for the simulation of these STS dynamics is based on comparison of observed simulation of STS dynamics to the specified target probabilities of occurrence and distributions.

Table 10. Types of Information Content Quality Degradation Simulated Through RPA-based Approach with Target Probabilities of Occurrence.

Type of Data	Error Type	Probability of Occurrence
Supply	Completeness: Fail to enter a new status	4%
	Accuracy: Erroneously add an extra digit (e.g., 0)	4%
	Precision: Round the value to the nearest 100	3% (for quantities > 500)

While the dynamics specified here were developed in coordination with the research sponsor, with the exception of the linear and bimodal distributions, it is important to note that they are not based on any empirical evidence. This research is concerned only with simulating specified dynamics in the exchange of information between simulations and ISs, not the identification of the appropriate dynamics themselves. Future work is needed to identify methods for determining appropriate distributions for simulation of temporal and information content quality degradation dynamics in support of sim-IS environments for training and analysis. Obstacles to this task include the lack of real-world military logistics operations for observation, inaccessibility of such operations and difficulty of measuring temporal information exchange dynamics in fluid military operations, limited representation of logistics ISs in staff training environments, and the limitations of field training exercises in terms of time and space which are necessary for stressing logistics command and control. One way of mitigating these obstacles is to develop collection plans and processes for use by uniformed researchers in future military operations. Sim-IS environments themselves may also serve to mitigate some of these factors and support a virtuous cycle where they provide insight regarding anticipated temporal dynamics as logistics ISs, processes, and personnel competencies coevolve. Ultimately, the sim-IS environments need to be instrumented to collect such information to provide lessons learned for training audiences and support the continuous improvement for both the JCSs being exercised and the RPA-based sim-IS environments themselves.

The verification of the RPA-based sim-IS architecture is limited to verification of sim-IS information exchange and the simulation of the specified dynamics, with an understanding that extreme cases either do not exist or are outside the scope of this research. Exception handling is not addressed, as the development of a prototype with production quality code is outside the scope of this research. While the misalignment of StartEx data is not the focus of this research (this is a problem for many distributed simulation environments (Bowers et al., 2011)), parametric data alignment for RPA-based information exchange does present several unique, important considerations relative to parametric data alignment for conventional simulations. A process was therefore developed and documented (Appendix D) to guide the alignment of parametric data across a JTDS scenario, simulation (JCATS), and logistics IS in support of RPA-based sim-IS environments. Some of the unique considerations identified in the development of this process, and recommendations for addressing them in the DSEEP, are discussed in Chapter VII.

2. Quantitative Measures and Thresholds for Verification of Sim-IS Information Exchange and Simulation of Information Exchange Dynamics

Verification of the proposed architecture for sim-IS information exchange includes verification of sim-IS information exchange itself as well as verification of the simulation of the specified information exchange dynamics. The former consists of determining whether the RPA-based sim-IS architecture supports information exchange with sufficient levels of accuracy, precision, and timeliness (H3). The latter consists of verifying the simulation of temporal dynamics (latency and timeliness) (H1) and the simulation of information content degradation (degraded accuracy, precision, and completeness) (H2). For H1 and H2, verification that the RPA middleware adequately simulates the specified dynamics in accordance with specified probabilities (for information content degradation) and distributions (for temporal dynamics) is divided between two parts: 1) generation of an RPA schedule which adequately simulates the probabilities or distributions for the respective dynamics and 2) execution of sim-IS information exchange in accordance with a given RPA schedule. This section describes the measures and statistical techniques used for verification of the RPA-based architecture for sim-IS information exchange and the simulation of information exchange dynamics.

a. Verifying Sim-IS Information Exchange

This research explores how an RPA-based "outside-in" approach to sim-IS information exchange can facilitate automated exchange of information between simulations and HitL ISs in support of exercises (research question 2), by verifying if RPA-based sim-IS information exchange can be conducted with sufficient accuracy, precision, and timeliness to support staff training (H3). Sufficient accuracy is defined as observing unintentional information exchange errors in fewer than 0.1 percent of sim-IS information exchange events. Sufficient precision is defined as observing unintentional information exchange precision errors (where information exchange values differ by a value of greater than an increment of 1 from the appropriate, translated status values) in fewer than 0.1 percent of sim-IS information exchange events.

Verifying sufficient timeliness in the execution of sim-IS information exchanges is a more complex issue, as what constitutes sufficient timeliness for a particular sim-IS environment is directly related to the number of sim-IS events to be executed, the latency and timeliness distributions to be simulated, and the number of RPA instances which can be supported. Rather than attempt quantitative verification that the RPA-based approach to sim-IS information exchange supports sufficient information exchange timeliness in such a general sense, two more valuable questions are determined to be:

1. Is there a way for sim-IS designers to determine the number of units, entities, and supply types supportable per RPA bot instance, based on the simulation and IS of a given sim-IS environment and the target timeliness and latency distributions?
2. How can the conceptual models for simulation and IS workflows be annotated to provide sim-IS environment designers the requisite information to conduct the calculation described in the previous question?

b. *Verifying Simulation of STS Dynamics: Temporal Dynamics*

This research explores how an RPA-based "outside-in" approach to sim-IS information exchange simulate temporal and content quality degradation of information across sociotechnical systems (research question 1) by verifying if RPA-based sim-IS information exchange can simulate temporal dynamics (H1) and information degradation dynamics (H2) which are sufficient to support staff training. Verifying the simulation of temporal dynamics is addressed in two parts: 1) verifying that timeliness and latency dynamics are adequately simulated in the RPA schedules generated by the RPA workflow and 2) verifying that RPA-based sim-IS information exchanges are executed in accordance with specified RPA schedules.

The simulation of temporal dynamics is determined by verifying a sufficient degree of alignment between the latency and timeliness variability distributions manifesting in RPA-generated RPA schedules and the target distributions defined previously. The chi square goodness of fit test is used to verify simulation of latency and timeliness variability distributions, with a significance level of 0.95. The threshold for rejecting the null hypotheses regarding scheduling temporal dynamics for simulation is therefore for the observed latency and timeliness (H1) distributions in RPA schedules generated by the RPA middleware to align with the target distributions with chi square goodness of fit test p-value > 0.95 for all scenarios.

Verification that RPA-based sim-IS information exchanges are executed in accordance with specified RPA schedules is supported by controlled simulation experiments conducted in the MOVES simulation lab. The threshold for rejecting the null hypothesis regarding simulation of temporal dynamics is for at least 99 percent of simulated RPA-based sim-IS extract and entry events to align with the extract and entry event times specified in the RPA schedule.

c. *Verifying Simulation of STS Dynamics: Information Content Quality Degradation*

As with the verification of temporal dynamics, the simulation of STS dynamics is conducted in two parts: 1) verification that scheduled dynamics align with the specified probabilities of occurrence and 2) verification that the information content quality degradation occurs as specified in the RPA schedule during the sim-IS information exchange events. Unlike the temporal dynamics, the scheduling of information content quality degradation occurs during the simulation itself, immediately following the extraction of information from the simulation.

Verifying simulation of information content quality degradation (H2) is conducted for three different dynamics: simulation of degraded accuracy, precision, and completeness. Verification of the scheduling of information content quality dynamics is defined in terms of the degree of alignment between the probability of occurrence in the RPA schedule and the target probabilities defined previously. A two-sided binomial test is used to verify simulation of information content quality degradation, with a significance level of 0.95. The threshold for rejecting the null hypothesis for H2 is for the frequency of the specified dynamics to align with the target probabilities with a p-value > .95 for each type of information content degradation simulated. Verification of the actual simulation of information content quality degradation during sim-IS information exchange is achieved by comparing the outcomes of the simulations with the associated RPA schedules and supply status tracker files. The threshold for rejecting the null hypothesis regarding simulation of information content quality degradation dynamics is the accurate simulation of effects for at least 99 percent of information exchange degradation effects specified in the RPA schedules and associated supply status tracker files.

3. Sim-IS Environment Simulation and Data Collection

This section discusses the experimental constructs employed for exercising the RPA middleware in the proposed sim-IS architecture to test the hypotheses as described previously. Two types of experimental constructs were employed for the verification process. Verification that the proposed RPA middleware can generate RPA schedules that adequately align with target temporal distributions and probabilities of occurrence, the first parts of H1 and H2, respectively, was supported by testing the RPA middleware in a standalone setup. The RPA middleware was exercised with each of the sets of parameters for temporal distributions and information content degradation probabilities presented previously. The RPA schedules generated under each of these scenarios were then evaluated using the previously described statistical techniques to verify representation of temporal distributions and information content degradation probabilities of occurrence in the resulting RPA schedules.

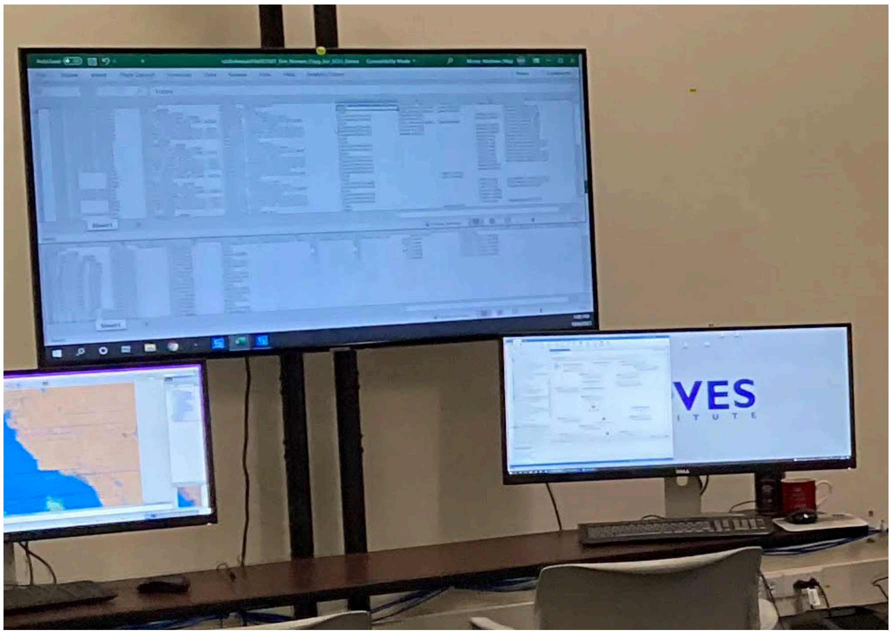
Verification that the proposed architecture supports sim-IS information exchange (H3) and the simulation of temporal and information content degradation dynamics specified in given RPA schedules (the second parts of H1 and H2) required a controlled sim-IS environment where the full process was exercised. The experimental control and data collection process required for this construct included imposing scheduled status changes on simulated units and entities, documenting those statuses at the scheduled data extraction times, documenting the values entered into the IS (CLC2S), and documenting the time of entry. Like the sim-IS data exchanges conducted manually by pucksters, this verification of information exchange dynamics required a tedious, manual process.

A scenario was developed in JTDS for use in both the JCATS-CLC2S and MTWS-CLC2S simulation environments. This scenario was developed using notional Marine Corps units with personnel, equipment, and supplies. The OBS was modified, and the requisite CLC2S StartEx files and initial RPA schedule were generated, through use of the BFG and processes described in Chapter V and Appendix D. For the MTWS-CLC2S and JCATS-CLC2S environments, all StartEx files were loaded in the CLC2S training server, simulation, and the RPA (UiPath) system prior to initialization and execution of the simulation and RPA workflow. Loading the simulation OBS and CLC2S feed files prior to each simulation ensured the supply levels and entity statuses were reset prior to each simulation scenario.

RPA schedules which approximated the latency and timeliness distribution specified previously for each scenario were loaded in the RPA system (UiPath) prior to execution of each simulation scenario. Lastly, the RPA schedule file produced by the BFG was trimmed to a supportable number of data extract and entry events for each scenario, based on the temporal distributions and anticipated extract and entry times. Before addressing the multiple factors influencing what constitutes a supportable number of data extract and events, a brief description of the verification process itself is necessary.

Verifying the simulation information is extracted and entered in the IS in accordance with the times specified in the RPA schedule requires documentation of the times when the requisite information is extracted and the times of entry into the appropriate IS. Verifying information content degradation for sim-IS information exchange requires documentation of the supply and entity statuses within the simulation when they are extracted and the values entered into the IS. The actions taken by the RPA workflow are too fast for the human eye to visually observe and document these four pieces of information. Instead, the times and values for data extracted from the simulation in each sim-IS scenario can be determined by modifying the supply level and entity readiness status values within the simulation immediately before and after each data extraction event is scheduled to occur. By comparing the known values at the data extraction times with the extracted simulation information within the RPA records, and the data subsequently entered into the IS, the time of data extraction can be determined. For example, if the fuel and ammunition statuses for a combat logistics battalion are scheduled to be extracted at 0900, the values associated with that unit are incremented immediately prior to 0900 and at 1000. If the values entered into the IS by the RPA workflow equal the values known to have existed in the simulation between 0900 and 1000, then the researcher knows the time (the hour window in this construct) when the data extraction occurred. Documenting the times and values associated with data entry into CLC2S does not require any such manual action by the researcher, as CLC2S maintains a record of the times when supply and equipment statuses are modified.

The researcher's modification of simulation data was guided by a copy of the RPA schedule for each simulation scenario, which also facilitated documentation of values to support analysis following the simulation. The process for manual modification of simulation data during the simulations is different depending on the simulation used in the scenario. Modification of entity readiness statuses and supply levels within MTWS can be executed during a simulation event by executing MTWS "batch file" scripts built in advance of the simulation scenario based on the RPA schedule. JCATS, however, does not provide such an option for running a script for modifying entity and supply statuses. Instead modifying JCATS supply and entity statuses requires the researcher to manually change the statuses of every entity and supply item in the JCATS GUI. Management of the MTWS-CLC2S and JCATS-CLC2S simulation environments was conducted in the MOVES lab, as seen in Figure 49.



Left Screen: JCATS simulation, where researcher manually modifies unit supply levels and entity readiness levels immediately before and after the times specified in the RPA schedule. Center Screen: Copy of the RPA Schedule, where the researcher documents the supply levels and entity readiness values before, during, and after the specified extraction time. Right Screen: Here the RPA workflow interacts with the simulation reports and CLC2S GUI.

Figure 49. Experiment Setup: JCATS-CLC2S Environment.

This difference made the MTWS-CLC2S environments more time efficient per unit, facilitating the simulation of all requisite units in a single day for each set of latency and timeliness distributions, while four days were necessary in the JCATS-CLC2S environments. For this reason, the MTWS-CLC2S environment was selected for simulation of additional distributions to ensure the tunability of the architecture for simulating additional latency and timeliness distributions.

When the initial RPA schedule is generated by the BFG application, all units are included. For each simulation scenario, this initial RPA schedule must be trimmed to include only those units, and supply items and entities (personnel and equipment) per unit, for which automated sim-IS information exchange is desired. Selection of these items and entities depends on multiple factors. In addition to exercise design considerations, a primary concern is ensuring the data extract and entry events are supportable with the number of RPA bot instances available. Every event takes some amount of time, for actions external and internal to the RPA workflow, and determining the supportable number of data extract and entry events per RPA bot instance depends on the times associated with those events as well as the target latency and timeliness distributions to be simulated. This question of how many data extract and entry events to include in a scenario is important for both the design of the verification experiments and the design of RPA-based sim-IS environments. It also directly addresses the third part of Hypothesis 3: identifying the timeliness of RPA-based sim-IS information exchange. A technique which was developed for addressing this issue, and its implications for RPA-based sim-IS design and development are addressed in the following section.

4. Determining a Supportable Number of Data Entry and Extract Events per Hour

For the JCATS-CLC2S scenario, the limiting factor for the number of sim-IS extraction and entry events is an artifact of the experimental design. Controlled simulation of the RPA-based sim-IS environment requires the manual adjustment of unit supply statuses throughout the execution of each simulation scenario. For MTWS, the manipulation of supply statuses can be conducted quickly with the execution of pre-generated scripts, called batch files. For JCATS, however, changes to supply statuses cannot be pre-scripted, and instead require a time consuming, tedious process. For the sim-IS environments simulated, the number of sim-IS information extract and entry events in the JCATS-CLC2S scenarios is therefore limited by the time requirements for the researcher's manual actions.

For the MTWS-CLC2S environment, the number of sim-IS information extract and entry events which can be included is limited by the speed of the RPA workflow in accomplishing all scheduled data extract and entry events for each hour. Designing the MTWS-CLC2S simulation environment therefore provided an opportunity to address the timeliness issues related to Hypothesis 3. To increase the time efficiency of this research, a technique was developed for approximating the number of data extract and entry events which could be supported in each simulation day by a single RPA bot instance. Sim-IS environment designers using RPA-based approach for sim-IS information exchange are expected to require a similar approach for determining how many RPA bot instances are required to support the planned number of daily supply and entity status updates for the requisite latency and timeliness distributions. This section provides an overview of the approach developed in this research while consideration of this issue in the context of DSEEP is discussed in Chapter VII.

Estimating the number of RPA extract and entry events supportable for a given sim-IS environment is a three-step process. The process is described here using information from the MTWS-CLC2S sim-IS environment scenario conducted using linear latency and timeliness distributions and a LogStat submission deadline of 1400. First the average speed of RPA-based information extract and entry times must be determined for a given simulation and IS pair. Ideally, this information would be included in RPA workflow documentation or sim-IS environment documentation (e.g., Federation Engineering Agreements Template [FEAT]). If such documentation is not available, the extract and entry speeds must be calculated by the sim-IS environment developer through experimentation. Even if documentation is available regarding speeds of RPA workflows with associated simulations and ISs, it would behoove sim-IS environment developers to conduct local tests, as network and processing speeds may differ between locations. For the MTWS-CLC2S scenario used in this discussion, which also included the exchange of only information for two supply items (fuel and rations) and no entities, the speed of information extract (from MTWS) ranged from 21.6 to 33.0 seconds, with an average of 27.0 seconds. The speed of information entry (into CLC2S) ranged from 27.2 to 57.0 seconds, with an average of 40.6 seconds.

A linear relationship was observed between the speed of both information extract events (Figure 50) and information entry events (Figure 51) and the number of events which have occurred previously. Extract and entry events can generally be expected to take increasingly longer as an RPA-based sim-IS environment simulation progresses over time.

This may be due in part to the design of the RPA workflow, as the size of the RPA schedule file increases as the RPA workflow progresses through the events. It may also be due in part to the impact of RPA workflows on computer memory, as some improvements in RPA extract and entry event speeds were observed when the machine hosting RPA workflows was restarted prior to each simulation scenario. The speeds of RPA entry events are impacted by multiple factors, including simulation of information content degradation dynamics. The fast entry event outliers, Group 1 in Figure 51, include 12 of the 13 units where entry of supply statuses was intentionally missed for one of the two supply types. The slow entry event outliers, Group 2 in Figure 51, with speeds of 55.6, 55.8, and 57 seconds, are the entry events which occur immediately following extract events. For this research, the average speeds of extract and entry events (27.0 and 40.6 seconds respectively) were used to estimate the supportable numbers of RPA instances, though future research should identify a method for considering the effects described previously.

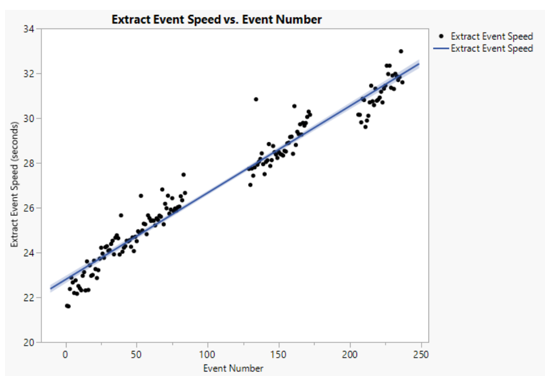


Figure 50. Speed of MTWS-CLC2S RPA Extract Events (in Seconds) Relative to Their Time of Execution.

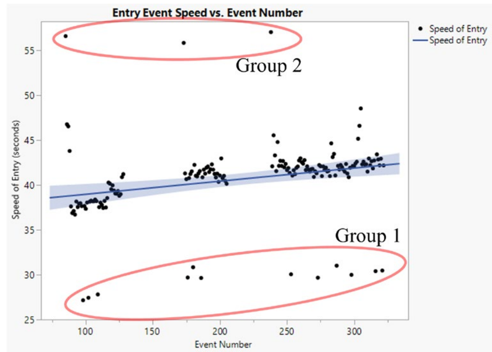


Figure 51. Speed of MTWS-CLC2S RPA Entry Events (in Seconds) Relative to Their Time of Execution.

Next, the sim-IS environment developer must identify the anticipated number of information entry and extract events per hour, based on the joint distribution generated from the specified latency and timeliness distributions and the LogStat submission deadline around which the timeliness variability distribution is applied. This can be calculated as a percentage of the total number of information entry and extract events for a given day, as demonstrated in Table 11.

Table 11. The Anticipated Number of Extract and Entry Events per Hour as a Function of "n" Units, Given a Joint Distribution of Latency and Timeliness Probability Distributions.

Target Timeliness Distribution:			Extract Time												
Timeliness (Hr)	Timeliness Probability (%)	Entry Time	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
-3	0.07	1400	0.0049n	0.0077n	0.0112n	0.0126n	0.0168n	0.0119n	0.0049n						
-2	0.08	1500		0.0056n	0.0088n	0.0128n	0.0144n	0.0192n	0.0136n	0.0056n					
-1	0.17	1600			0.0119n	0.0187n	0.0272n	0.0306n	0.0408n	0.0289n	0.0119n				
0	0.26	1700				0.0182n	0.0286n	0.0416n	0.0468n	0.0624n	0.0442n	0.0182n			
1	0.18	1800					0.0126n	0.0198n	0.0288n	0.0324n	0.0432n	0.0306n	0.0126n		
2	0.14	1900						0.0098n	0.0154n	0.0224n	0.0252n	0.0336n	0.0238n	0.0098n	
3	0.1	2000							0.007n	0.011n	0.016n	0.018n	0.024n	0.017n	0.007n
Target Latency Distribution:				Latency Probability (%):			0.07	0.11	0.16	0.18	0.24	0.17	0.07
				Latency (Hr):			6	5	4	3	2	1	0

The maximum number of times when any particular pair of information extract and entry values can occur is determined by multiplying the percentage by the total number of units included in a scenario (n) and rounding up to the next whole number. For example, for a scenario where RPA-based sim-IS information exchanges will be executed for 1,000 units, data extract at 13:00 and entry at 14:00 can be expected to occur for a maximum of 12 units. The maximum number of extract and entry events for any given hour in the day is determined by summing up the respective columns and rows, as seen in Table 12, which demonstrates that the largest number of extract and entry events that can be expected to occur in any given hour for this scenario is 364 (occurring at 17:00, with a maximum of 262 information entry events and a maximum of 102 information extract events).

Table 12. The Maximum Number of Information Extract and Entry Events per Hour for n = 1000.

Target Timeliness Distribution:				Extract Time												
Timeliness (Hr)	Timeliness Probability (%)	Entry Time	Maximum Entry Events per Hour:	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
-3	0.07	1400	72	5	8	12	13	17	12	5						
-2	0.08	1500	83		6	9	13	15	20	14	6					
-1	0.17	1600	171			12	19	27	31	41	29	12				
0	0.26	1700	262				19	29	42	47	62	44	19			
1	0.18	1800	182					13	20	29	33	43	31	13		
2	0.14	1900	142						10	16	23	25	34	24	10	
3	0.1	2000	100							7	11	16	18	24	17	7
Maximum Extract Events per Hour:				5	14	33	64	101	135	159	164	140	102	61	27	7
Maximum Events (Entry + Extract) per Hour:				5	14	33	64	101	135	231	247	311	**364**	243	169	107
Target Latency Distribution:					Latency Probability (%):					0.07	0.11	0.16	0.18	0.24	0.17	0.07
					Latency (Hr):					6	5	4	3	2	1	0

The third and final step is to apply the extract speed ($t_{extract}$) and entry speed (t_{entry}) values to these distributions, to identify the largest number of events which can be conducted without exceeding the desired time windows for the events. In the scenarios used for this research, extract and entry events occur within one-hour time windows, with all extracted supply statuses entered within the same day. The highest number of events (N) which can be supported by an individual RPA bot can be determined for each hour by solving

$$\text{ceiling}(N \times P_{extract}) \times t_{extract} + \text{ceiling}(N \times P_{entry}) \times t_{entry} < 1 \text{ hour},$$

where $P_{extract}$ and P_{entry} are the percent of n extract and entry events, respectively, which are anticipated for a given hour based on the joint distribution for temporal dynamics.

5. Exception Handling

Although exception handling is outside the scope of this research, a few common issues were identified which have implications for the design of RPA workflows for RPA-based sim-IS environments. These consisted of account timeout and unscheduled outages in the CLC2S training server. While exception handling for account timeout can be addressed by designing the RPA workflow to log in as required, the inaccessibility of the simulation or IS interfaces requires human intervention.

How an RPA workflow is designed to handle account timeout depends on the level of risk organizations are willing to take regarding automated RPA access to ISs. While some organizations have begun employing unattended RPA bots, other organizations do not. Those organizations which employ unattended RPA bots may only authorize unattended RPA bots for certain workflows and ISs. One consideration when determining whether to employ RPA in an attended or unattended manner is an organization's willingness to save account usernames and passwords for the RPA both to automatically enter when accessing a password protected IS. If an organization is not comfortable with storing usernames and passwords on their machines for RPA bot use, a human operator may be required for facilitating initial RPA bot access to an IS and to facilitate subsequent login when account timeout occurs.

While the workflows developed in support of this research were designed in just such an unattended fashion, a work around was developed which prevented frequent account timeouts from occurring. By designing the RPA workflow to refresh the user interface at regular, frequent intervals, account timeout occurrences were greatly diminished. This is not a guaranteed method, however, and still requires the presence of a human operator to oversee the process and step in to log into the account should a timeout occur.

Network and server availability present a problem, regardless of whether sim-IS information exchange is conducted manually or through RPA. If the interface for a simulation or IS is unavailable, RPA workflow processes must be able to pick up where they left off once the simulation or IS server becomes operational. For the RPA workflows designed in support of this research, this was accomplished by continuously saving the RPA files to the local machine throughout the execution of the simulations. With this approach, when RPA workflows are restarted following the return of services, the workflow picks up with the schedule and requisite information from where the process was interrupted.

C. TEMPORAL DYNAMICS SIMULATION RESULTS

This section presents results for verification of the proposed RPA middleware and sim-IS architecture for the simulation of temporal dynamics (H1). As will be seen, all chi square goodness of fit tests for H1, part 1 resulted in p-values > 0.95 while 100% of the observed sim-IS information exchange extract and entry events for H1, part 2 occurred in accordance with the entry and extract times specified in the RPA schedules. As all observed latency and timeliness distributions align with target distributions with chi-square goodness of fit p-values > 0.95, and the observed proportion of sim-IS information exchange events with accurate times of occurrence is > 0.99 for all scenarios tested for H1, part 2, we reject the null hypotheses that simulated latency and timeliness distributions do not align with the target distributions.

1. H1, Part 1: Verifying Latency and Timeliness Variability Distributions in RPA Schedule Generation

For the first part of H1 testing, the alignment of latency and timeliness distributions with the target distributions was tested using the chi-square goodness of fit test (alpha = 0.95). For the simulation of unimodal latency and timeliness distributions, observed distributions in the generated RPA schedule aligned with the target distributions with chi square goodness of fit p-values of 0.9748 and 0.9968 respectively, as seen in Figure 52. The observed joint distribution for the unimodal distributions aligned with the target joint distribution with a chi square goodness of fit p-value of 1.00. It is reasonable for the observed joint distribution to align with the target joint distribution so closely, as the algorithm leveraged for simulating target latency and timeliness distributions is designed to ensure simulation of the joint distribution as opposed to the simulation of either the latency or timeliness distributions alone.

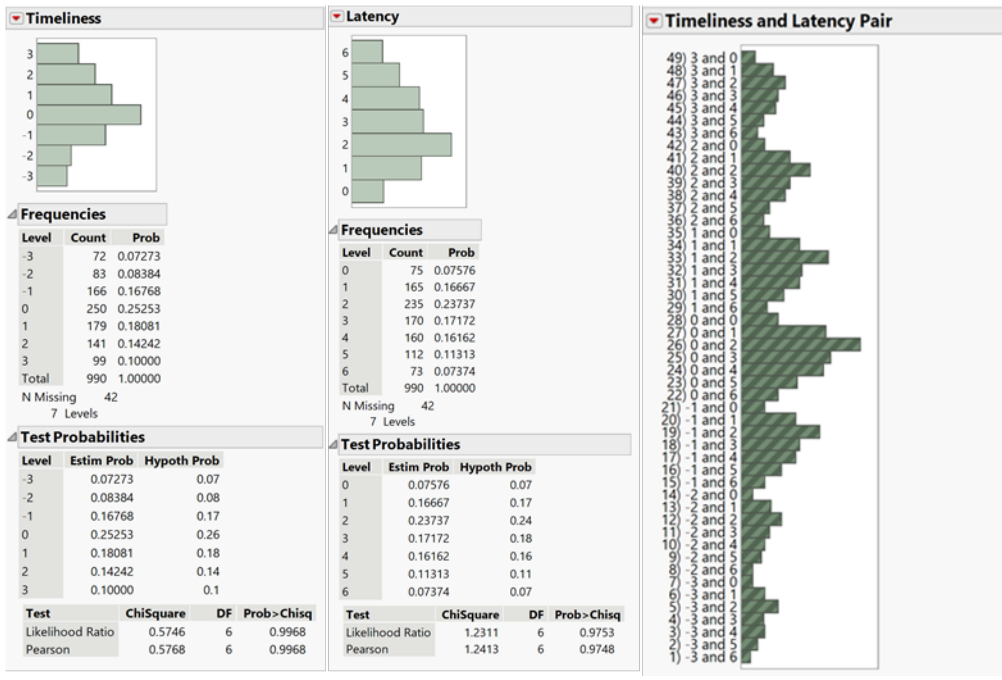


Figure 52. Results for RPA Schedule Generation with Unimodal Latency and Timeliness Distributions.

Similar results were observed for the bimodal and linear distributions. For the simulation of bimodal latency and timeliness distributions, observed distributions in the generated RPA schedule aligned with the target distributions with chi square goodness of fit p-values of 0.9574 and 0.9865 respectively, as seen in Figure 53. The observed joint distribution for the bimodal distributions aligned with the target joint distribution with a chi square goodness of fit p-value of 1.00. For the simulation of linear latency and timeliness distributions, observed distributions in the generated RPA schedule aligned with the target distributions with chi square goodness of fit p-values of 0.9859 and 0.9986 respectively, as seen in Figure 54. The observed joint distribution for the linear distributions aligned with the target joint distribution with a chi square goodness of fit p-value of 1.00.

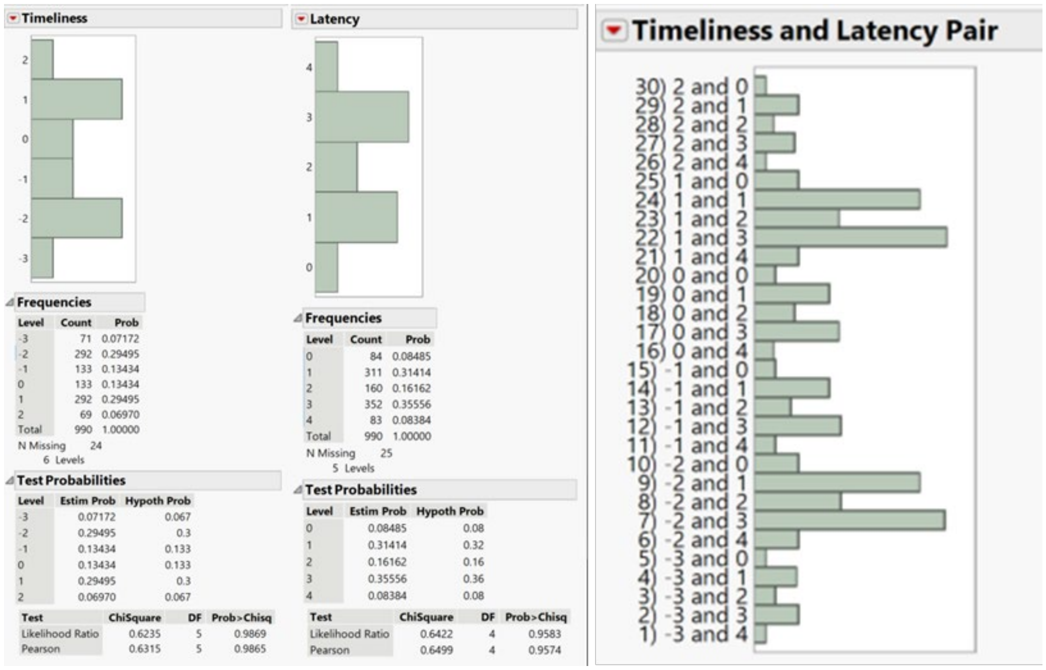


Figure 53. Results for RPA Schedule Generation with Bimodal Latency and Timeliness Distributions.

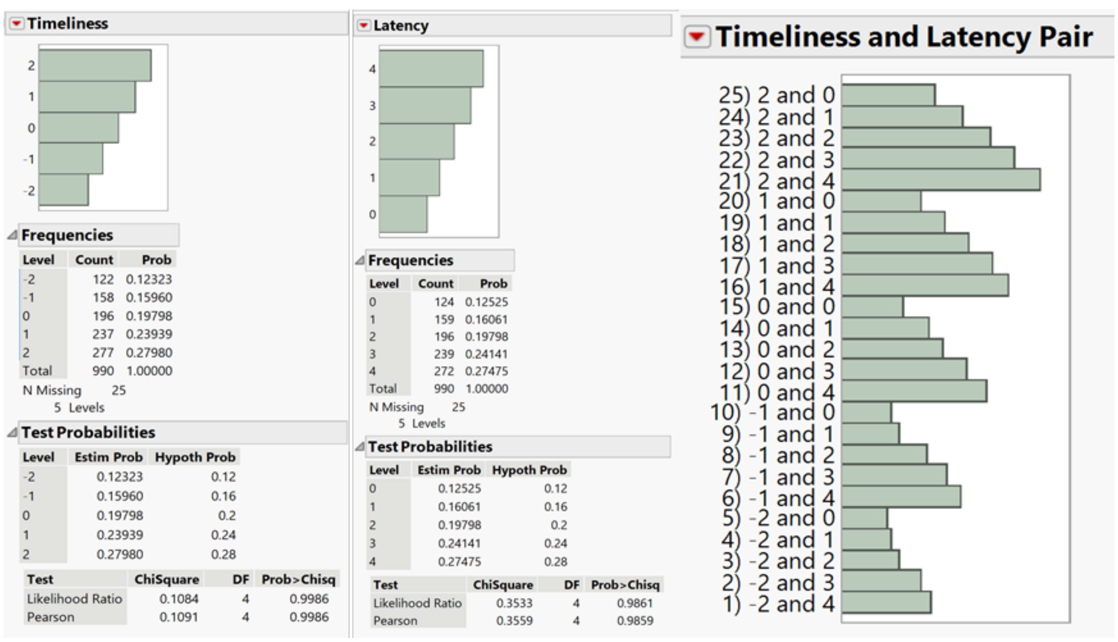


Figure 54. Results for RPA Schedule Generation with Linear Latency and Timeliness Distributions.

2. H1, Part 2: Verifying Simulation of Temporal Dynamics Specified in the RPA Schedule

For the second part of H1 testing, simulation of temporal dynamics through the proposed RPA-based sim-IS information exchange architecture was tested using JCATS-CLC2S and MTWS-CLC2S environments. Tests using the JCATS-CLC2S environment were conducted using unimodal latency and timeliness distributions across four different days of execution with a total of 128 simulated units. Tests using the MTWS-CLC2S environment were conducted using unimodal, bimodal, and linear latency and timeliness distributions, with one day of simulation each and 197, 165, and 165 simulated units respectively. With 100% alignment observed in all scenarios, as specified in Table 13, these tests support verification that the timing of RPA-based extraction of information from simulations and entry of information into appropriate ISs is executed by the RPA-based sim-IS architecture in accordance with the extract and entry times specified in the RPA Schedule.

Table 13. Observed Alignment of Sim-IS Information Exchange Extract and Entry Event Timing with Times Specified in RPA Schedules.

JCATS-CLC2S Unimodal Distributions	MTWS-CLC2S Unimodal Distributions	MTWS-CLC2S Bimodal Distributions	MTWS-CLC2S Linear Distributions
128/128 (100%)	197/197 (100%)	165/165 (100%)	165/165 (100%)

D. INFORMATION CONTENT QUALITY DEGRADATION SIMULATION RESULTS

This section presents results for verification of the proposed RPA middleware and sim-IS architecture for the simulation of information content quality degradation dynamics (H2). As will be seen, the probabilities of error occurrence for H2, part 1 were inadequately aligned with the target probabilities of occurrence to achieve statistical significance, with all two-sided binomial test p-values < 0.95. 100% of the observed information content degradation events occurred correctly in the simulated RPA-based sim-IS information exchange JCATS-CLC2S and MTWS-CLC2S scenarios, in accordance with the associated RPA supply and schedule files. As all observed probabilities of error occurrence for H2, part 1 fell below the 0.95 threshold, we fail to reject the null hypotheses that RPA-based sim-IS information exchange architecture cannot support simulation of information content quality degradation with specified probabilities of occurrence.

1. H2, Part 1: Verifying Probabilities of Occurrence for Information Content Quality Degradation Events in RPA Supply File Generation

For the first part of H2 testing, the alignment of information accuracy, precision, and completeness errors' probabilities of occurrence with the target probabilities was tested using two-sided binomial tests. For all three error types, while the target probabilities fell within the observed 95% confidence intervals, the observed probabilities of occurrence in the generated RPA supply status file did not reach the 0.95 threshold for rejecting the null hypothesis with statistical significance. These results, presented in Table 14, suggest that the proposed RPA-based sim-IS information exchange architecture supports the simulation of information content quality degradation with specified probabilities of occurrence with practical significance but not statistical significance (Kirk, 1996).

Table 14. Results for Scheduling Simulation of Information Content Degradation Error Occurrence.

Information Exchange Error Type	Number of Simulated Supply Item Instances	Target Probability of Error Occurrence	Observed Probability of Error Occurrence	p-value for Two-Sided Binomial Test	95% Confidence Interval
Degraded Accuracy (Added a digit)	1000	4.0%	4.3%	0.6277	3.13% to 5.75%
Degraded Precision (Round to nearest 100)	1000	3.0%	3.2%	0.7101	2.20% to 4.49%
Degraded Completeness (Missed Entry)	1000	4%	3.7%	0.6868	2.62% to 5.06%

These limitations of the RPA-based middleware in adequately representing the specified probabilities of error occurrence are likely due to the limitations of the RPA software's organic pseudo random number generator discussed in Chapter IV. While the effects of this pseudo random number generator limitation are mitigated for the simulation of temporal distributions through the algorithm presented in Chapter IV, the algorithm cannot be applied to selection of information errors with the current architecture. This is because, unlike the scheduling of extract and entry events, which occurs prior to an exercise, the application of errors occurs during the execution of the sim-IS environment. Attempting to apply this algorithm for information content quality degradation events would result in a disproportionate number of errors being applied to those units with information extract events occurring later in the simulation day.

2. H2, Part 2: Verifying Simulation of Information Content Quality Degradation Events in Sim-IS Information Exchange Scenarios

For the second part of H2 testing, simulation of information content quality degradation through the proposed RPA-based sim-IS information exchange architecture was tested using JCATS-CLC2S and MTWS-CLC2S environments. Tests using the JCATS-CLC2S environment were conducted across four different days of simulation with a total of 672 sim-IS information exchanges for simulated supply item statuses. Tests using the MTWS-CLC2S environment were conducted across three days of simulation with a total of 916 sim-IS information exchanges for simulated supply item statuses. With 100% alignment observed in all scenarios, as specified in Table 15, these tests support verification that information content quality degradation events are executed by the proposed RPA-based sim-IS information exchange architecture in accordance with the specified RPA supply and schedule files.

Table 15. Observed Simulation of Specified Information Content Quality Degradation Events (# of Events Observed Occurring Appropriately / # of Events Scheduled).

Error Type	JCATS-CLC2S (672 Supply Item Statuses Exchanged)	MTWS-CLC2S (916 Supply Item Statuses Exchanged [435 for rounding errors])
Accuracy (Adding a Digit)	22 / 22 (100%)	28/28 (100%)
Precision (Rounding Value)	15/15 (100%)	11/11 (100%)
Completeness (Missing Entry)	31/31 (100%)	35/35 (100%)

E. RPA-BASED SIM-IS INFORMATION EXCHANGE RESULTS

This section presents results for verification of the proposed RPA middleware and sim-IS architecture as an approach for automating sim-IS information exchange with sufficient accuracy, precision, and timeliness to support staff training (H3). Accuracy and precision of sim-IS information exchange were tested through four JCATS-CLC2S simulations (totaling 672 sim-IS information exchanges for simulated supply items) and three MTWS-CLC2S simulations (totaling 916 sim-IS information exchanges for simulated supply items). With regard to sim-IS information exchange accuracy, for both MTWS-CLC2S and JCATS-CLC2S sim-IS environments simulated, no unintentional data exchange errors were observed. With regard to precision, for both MTWS-CLC2S and JCATS-CLC2S sim-IS environments, all sim-IS information exchange events observed fell within an increment of one (1) of the appropriate value. The observed sim-IS information exchanges met the thresholds specified for sufficient accuracy (observing unintentional data exchange errors less than 0.1% of the time) and precision (all supply statuses entered into the appropriate IS with a value within an increment of 1 from the appropriate, translated status). Based on this observed performance of the sim-IS information exchanged architecture prototype across JCATS-CLC2S and MTWS-CLC2S environments, we find that the proposed sim-IS information exchange architecture supports sufficient accuracy and precision in sim-IS information exchanges to support staff training.

The question of whether RPA-based sim-IS information exchange is timely enough is a much richer question. For any given sim-IS environment, whether the RPA-based sim-IS information exchange approach is timely enough depends on the speed of information extract and entry events associated with the simulations and ISs to be included in that particular sim-IS environment. It also depends on the set of latency and timeliness distributions to be employed as well as the number of simulated units and the number of supply item instances for which statuses are to be exchanged for each simulated unit. The RPA-based sim-IS information exchange architecture is designed to support the allocation of units across multiple RPA bot instances, in case one RPA bot instance is insufficient for a given scenario. A process is required which identifies the requisite number of RPA bot instances for a given sim-IS environment and associated scenario and informs the allocation of units across RPA bot instances.

The first step in such a process is determining the maximum number of sim-IS information extract and entry events can be expected to occur for each hour window of a sim-IS exercise, based on a given joint distribution of temporal dynamics. Such a process was presented in Section B.4. The results presented in C.1 support the efficacy of this approach, as the RPA-generated RPA schedules consistently aligned with the target joint distributions with a chi square goodness of fit p-value of 1.00.

The second step requires a way of identifying the time required for executing the extract and entry events associated with the RPA workflows for the sim-IS environment's simulation(s) and IS(s). While Section B.4 identified the speed of extract and entry events for a MTWS-CLC2S environment, future work is necessary for evaluating how numerous factors influence the RPA workflow event speeds for different simulations and ISs and documenting them to inform the design and development of subsequent sim-IS environments.

An RPA-based approach for sim-IS information exchange is not suitable for all sim-IS information exchange requirements. Simulating the continuous, near real-time delivery of entity location updates to GPS systems, for example, is better suited for protocol-based sim-IS information exchange. The timeliness of the RPA-based sim-IS information exchange approach is one of several factors which must be considered when determining its use compared to the conventional sim-IS information exchange approaches (protocol-based exchange, pucksters, and engineering ad hoc point-to-point interoperability). Chapter VII presents some of these considerations as a spectrum to facilitate comparison of each approach's capabilities and limitations against sim-IS environment requirements.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RPA-BASED SIM-IS ARCHITECTURE VALIDATION

The evidence obtained from the verification of RPA-based sim-IS information exchange architecture for sim-IS information exchange and simulation of STS dynamics is necessary but not sufficient for validation of the proposed RPA-based sim-IS architecture. After identifying the overlap between quantitative verification and quantitative validation of the RPA-based sim-IS architecture, this chapter presents the results of qualitative validation efforts for the proposed RPA-based sim-IS architecture. DOD Instruction 5000.61 defines validation as “the process of determining the degree to which a model or simulation and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model” (USD(AT&L), 2009, p. 10). In addition to determining whether a simulation accurately represents an aspect of the real world, validation also addresses whether an M&S tool is “suitable for its intended use” (Appleget et al., 2010, p. 10) in how that aspect of the real world is to be simulated in an operational context. The RPA-based sim-IS environments proposed in this research are intended to automate sim-IS information exchange and simulate sociotechnical system dynamics like information latency and submission timeliness variability in support of staff training. The target sociotechnical systems and their associated information exchange dynamics are expected to continuously coevolve with all the elements of the associated JCSs (ISs, work environments, and user competencies).

When considering the first half of the definition of validation, the question of what constitutes “an accurate representation of the real world” presents a clear challenge for the simulation of continuously evolving sociotechnical systems and their associated dynamics. The referent for RPA-based sim-IS environments, the JCSs and associated sociotechnical system dynamics, are expected to continuously evolve. This is the nature of JCSs. Validation of a simulation for such a referent must be considered in light of the second half of the definition, with “the perspective of the intended uses of the model” guiding the validation process.

> In all cases, the purpose of the validation effort is to obtain evidence that the M&S is suitable for its intended use. That evidence can be obtained

> through a variety of evaluative quantitative and qualitative techniques, depending on the nature of the M&S, the particular use, and the level of risk that can be tolerated in the findings. (Appleget et al., 2010, p. 10)

Validation of the RPA-based sim-IS information exchange architecture is addressed here in three parts. First is validation of the RPA-based sim-IS information exchange architecture in the automation of sim-IS information exchange and the simulation of information exchange dynamics specified by the training environment designers. This is supported by the experimentation conducted in the verification process. Verification that the RPA-based sim-IS architecture adequately automates sim-IS information exchange and simulates specified STS dynamics provides evidence necessary for validation of achievement of user requirements by the RPA-based sim-IS architecture. This quantitative validation is important, but not sufficient, for the broader validation of the proposed architecture.

The second part is a qualitative analysis of the RPA-based sim-IS information exchange architecture with regard to its utility for its intended use: sim-IS information exchange in support of staff training exercises. This is supported by a field demonstration conducted aboard MCAGCC with the MCLOG M&S staff to validate the utility of the RPA-based sim-IS architecture in context. Structured interviews were conducted to receive subject matter expert feedback on the utility of the RPA-based sim-IS information exchange architecture for meeting the needs of sim-IS information exchange for their staff training exercises.

The third and final part of the validation process for the proposed RPA-based sim-IS architecture is identification of a process which can support the continuous validation of RPA-based sim-IS environments as all components of the referent evolve over time. It is important to recognize that validation is a process, and for the simulation of continuously coevolving referents validation must be approached as a continuous process.

A. FIELD DEMONSTRATIONS AND INTERVIEWS

From 18 to 22 October 2021, field demonstrations for the MTWS-CLC2S and JCATS-CLC2S prototypes were conducted aboard MCAGCC for representatives of the simulation staffs of MCLOG and the MCAGCC Battle Simulation Center (BSC). MCLOG and BSC M&S personnel observed these demonstrations and participated in multi-part structured interviews (see Appendix F for the list of interview questions) which addressed their existing processes for sim-IS integration and provided feedback on the performance of the demonstrated RPA middleware relative to their organizations' requirements for sim-IS environments. Prior to their execution, the demonstration plan and interview questions underwent a human subjects research determination review by the Naval Postgraduate School Institutional Review Board (IRB). The IRB found that the demonstration and interviews do not constitute human subjects research and therefore had no requirement for IRB approval. The interviewees uniformly reported that the demonstrated RPA based middleware met their organizations' requirements for sim-IS information exchange. Interviewees also stated that the architecture and RPA middleware present an advancement in sim-IS information exchange, in terms of both sim-IS interoperability and the simulation of STS dynamics.

1. Field Demonstration Timeline

Demonstrations were conducted within the MCLOG exercise control facility, using MTWS and JCATS instances maintained by the MCLOG simulation team and a UiPath instance setup on a MCLOG laptop. The demonstrations were divided between demonstrating the execution of RPA-based sim-IS architecture and demonstrating the sim-IS StartEx data generation and initialization process necessary for implementing the RPA-based sim-IS architecture. Following resolution of initial networking configuration issues, previously configured StartEx files were loaded into the MCLOG instances of the MTWS and JCATS simulations and locally hosted CLC2S training server.

The first day of the trip was spent addressing networking configurations, so that the simulations and the computer hosting UiPath software were positioned on the same network with access to the CLC2S training server. The UiPath Studio software is available in the Marine Corps Software Center on the Marine Corps Enterprise Network, making it available for installation by any Marine Corps organization with an account. The second day of the trip was spent loading scenario files for the demonstration. An initial obstacle was encountered with the MTWS simulation as the intended version of MTWS was unavailable and a different MTWS version (and different set of parametric data) had to be used in its place. This resulted in a limited number of supply classes being represented in the MTWS simulation (only the food, water, and fuel supply classes were present in the MTWS parametric data and none of the ammunition variants). While this obstacle did not degrade the simulation of the RPA-based sim-IS architecture functionality, it did identify the importance of StartEx data synchronization.

Demonstration of the execution and management of the RPA-based sim-IS information architecture was conducted with the MCLOG M&S staff on day 3 for both JCATS-CLC2S and MTWS-CLC2S environments. Demonstration of the JCATS-CLC2S scenario was repeated on the fourth and fifth days of the trip for two members of the MCAGCC BSC.

The sim-IS StartEx data generation process for RPA-based sim-IS environments was demonstrated for half of the MCLOG staff on the fourth day and for the other half of the staff on the fifth and final day. The StartEx data generation process included creation of an entirely new scenario, using a MCLOG JTDS scenario and synchronizing entity and supply class identifiers across JTDS, JCATS, and CLC2S parametric data. While the StartEx synchronization process is important for any sim-IS environment, this demonstration highlighted its particular importance for supporting the RPA-based sim-IS architecture and the unique considerations which need to be addressed for supporting RPA-based sim-IS information exchange. In the days during and following the StartEx data synchronization process demonstration, the process was documented on behalf of the MCLOG M&S staff. This documented sim-IS StartEx data generation process is presented in Appendix D.

2. Demonstration and Interview Results

As the Marine Corps' intermediate logistics schoolhouse, MCLOG is the primary stakeholder for the stimulation of Marine Corps logistics information systems through simulation-supported staff training exercises. Structured interviews conducted with three MCLOG simulations staff yielded subject matter expert validation of the RPA-based sim-

IS architecture as a means of automating sim-IS information exchange and simulation of STS dynamics which are infeasible with conventional sim-IS information exchange methods for Marine Corps logistics staff training exercises. The interviews also captured the state of MCLOG sim-IS environment design and execution processes as well as their desired future capabilities. Finally, the demonstration afforded a cursory comparison with the functionality provided by the recently fielded MTWS-CLC2S information exchange feature.

Following the demonstration of RPA-based sim-IS architecture execution and management, ***all MCLOG M&S staff affirmed that the demonstrated functionality meets their needs for sim-IS information exchange and that it would advance their capabilities with the simulation of STS dynamics***. Furthermore, all staff members agreed that the low-overhead automation of sim-IS information exchange would facilitate the introduction of additional logistics information systems into MCLOG simulation-supported staff training exercises that would otherwise be too expensive in manpower, time and/or engineering costs. Following demonstration of the StartEx data generation process required for RPA-based sim-IS environments, the MCLOG M&S staff unanimously found that the process presented improvements in exercise design and control but found that it would also require changes to existing processes. The following sections present the main points taken from the MCLOG M&S subject matter expert interviews and specific insights garnered from demonstration of the RPA-based sim-IS architecture in MCAGCC. Responses from interviewees are included in these sections, without attribution to individuals. Where the responses of multiple interviewees are presented regarding a particular topic, the responses are separated by asterisks.

a. Existing Sim-IS Information Exchange Processes and Requirements (Questions 1 and 2)

Initial questions addressed in the structured interviews were designed to elicit an understanding of current capabilities and limitations for sim-IS information exchange. The Marine Corps does not maintain standardized processes for the synchronization of sim-IS StartEx data or the execution of sim-IS information exchange in support of staff training exercises. Sim-IS information exchange processes, which are predominantly executed via pucksters for HitL ISs, are therefore heavily dependent on institutional knowledge in the form of personnel experience and local standard operating procedures. Furthermore, the processes for sim-IS information exchange are unique for different sets of simulations and operational ISs.

The MCLOG M&S section has a demonstrated history of integrating constructive simulations and logistics information systems in support of logistics staff training exercises (Morse, 2016, 2017). Despite this history and the noteworthy expertise of MCLOG's current simulation professionals, the section has had few opportunities in recent years to exercise its sim-IS information exchange processes. This has been largely due to the limitations imposed on live staff training exercises due to the coronavirus pandemic beginning in 2019. In recent years, the MCLOG M&S section has explored a transition from MTWS to JCATS as the primary constructive simulation for supporting local logistics staff training exercises. Although a tentative process has been developed to support the manual sim-IS information exchange between JCATS and CLC2S, this process and the associated excel-based tool have not been tested in a live exercise. The interviews therefore focused on documenting insights from senior members of the M&S section regarding the section's processes for sim-IS information exchange between MTWS, the CLC2S, and TCPT ISs.

When asked to describe the time and manpower requirements for MCLOG's sim-IS information exchange processes (question 1a), interviewees consistently described the processes as time and manpower intensive. The process for manual MTWS-CLC2S sim-IS information exchange was described as sometimes requiring "a minimum of four hours a night." The MCLOG M&S Section's efforts to develop a time and manpower-efficient sim-IS information exchange process for JCATS-CLC2S environments is ongoing. Initial effort resulted in time and manpower requirements similar to those experienced with MTWS-CLC2S environments

> Previously, this was done by a swivel-seat solution in which one individual pulled a JCATS report and read out supply numbers to someone who would manually input those numbers in CLC2S.TCPT. This tedious process required a minimum of 4 hours and was prone to human errors.

Efforts to augment JCATS-CLC2S information exchange with an excel-based tool, while untested, were described as being expected to yield minor to moderate time and manpower savings.

> This process still required a user to pull individual reports and burn them onto a CD to be exported. Once this was done a series of formulas would be ran that compiled and aggregated information into a format that could be consumed by CLC2S. This entire process still required an average of 2 hours.
>
> * * *
>
> In theory, the process of taking data out of JCATS and pushing it to CLC2S should take no longer than 10 minutes if the MCLOG-developed excel tool works properly. However, getting the tool to work properly and select the correct cells from JCATS reports is time intensive.

When asked to describe the possibility for unintentional human error in the manual exchange of data (question 1b), MCLOG and BSC staff unanimously reported a high risk of unintentional error resulting from their existing manual sim-IS information exchange processes.

> The unintentional human error was always an issue. One example, [for] Class I – bottled water vs. bulk; some folks would have different views [for one person] 1 case equals 12 bottles of water each at 16 oz...while another person would say...1 case equals 24 bottles of water each at 8 oz. In this example...if not clarified by exercise design...the staff would introduce error simply by which NSN was used or made up.
>
> * * *
>
> The possibility for unintentional human error in the manual exchange of data is very high.

One interviewee identified how unintentional errors encountered in existing sim-IS information exchange processes resulted in the training organization itself questioning whether the information systems should continue to be employed.

> Unintentional human error was common with the swivel seat solution. Operators would become careless as they rushed to input data or copy and paste data hundreds of times. It was fairly common to have users mistype, omit digits or simply paste information into the wrong locations. This

problem was so common a working group was assembled to identify if MCLOG should no longer utilize these systems [CLC2S and TCPT].

This effect of sim-IS information exchange shortcoming on impressions of the utility of the supported operational ISs mirrors an experience during the Marine Corps' LSE 2016. During LSE-16, a great disparity occurred between simulation ground truth data and the supply and entity statuses reflected in operational logistics ISs (Hensien, 2016). This disparity was the result of ineffective StartEx synchronization and dynamic sim-IS information exchange processes for the sim-IS environment. During the exercise, one of the unit commanders from the training audience misinterpreted this misalignment as reflecting limitations in the operational ISs rather than being due to design and management of the sim-IS environment.

When asked if existing sim-IS information exchange processes limit the number of operational ISs which could be included in logistics staff training exercises (question 1c), MCLOG M&S section personnel unanimously asserted that they do.

> Yes, it limits the number of other systems we could add to the mix for [a] full logistics picture.

* * *

> During the transition of data, a minimum of two MCLOG simulation operators are no longer available to conduct any other task. This shortfall would commonly lead to extended work hours reaching the 15 to 18 hour mark. This would also prevent the update of data during training hours as no simulation operator would be available to support any other task or the training audience.

* * *

> Absolutely. We have talked several times about incorporating real systems administration personnel utilities for exercises, but the common issue is determining who would be responsible for the installation and management of these systems.

The MCLOG M&S personnel were next asked how their organization provides training for the Training and Readiness Tasks LOG-OPS-7002/8002 "Receive and Validate Support Requests," in the context of simulation-supported staff training environments (question 2). The MCLOG M&S section has supported training for these T&R standards in the past by simulating support request and LogStat reporting errors through manual modification or deletion of information exchanged between the simulation(s) and information system(s). MCLOG has used simulated LogStat reporting errors to create a "domino effect," with the training audience experiencing second and third order effects if they failed to identify and appropriately respond to entity readiness and supply status reporting errors in CLC2S.

> MSEL [master scenario events list] events/injects are created that force the training audience to validate [support requests]…MCLOG events created specifically to force validation by the introduction of errors were only done via scripted injects and were planned on average only twice per exercise event. Anything more than this would become unwieldly as no mechanism to track the errors or actions taken as a result of those errors exists.

In recent years, multiple factors have resulted in a decreased representation of these errors. The transition from MTWS to JCATS and the staff turnover have resulted in decreased simulation of such errors, as the M&S section has had "a hard enough time just trying to get accurate data to and from the simulation to drive decision-making from the training audience." An additional reason identified in the interviews for decreased representation of errors in support requests is anecdotal evidence of decreased capability among the training audiences to leverage their logistics ISs even without simulated errors. The staff asserted that they would return to simulating LogStat reporting errors if the training audiences were better prepared to respond to them.

No standardized organizational system exists for imposing sociotechnical system dynamics like latency and timeliness variability in the information presented to the training audience to prepare them for such dynamics in the real-world use of their information systems. The training audiences unintentionally generate some information exchange errors themselves and the time intensive manual sim-IS information exchange process results in some limited latency, though this latency is not related to any specified latency distribution target to be simulated.

> A natural delay existed as response cell personnel would manually input data from a script or the simulation back into the exercise via CLC2S, emails or messaging applications.

b. Execution and Management of RPA-based Sim-IS Architecture

The second set of interview questions addressed the performance of the RPA-based sim-IS architecture for supporting the intended use for sim-IS environments. In response to question 3, ***all MCLOG M&S section personnel responded that the RPA-based sim-IS information exchange prototype meets their organization's requirements for the exchange of data between a simulation and CLC2S for their organization's staff training environments.***

> Absolutely. I cannot emphasize enough how much of a game-changer the technology and its implementation will be. Our processes will need to be refined and adapted to enable the use of the technology.
>
> * * *
>
> Absolutely. The RPA exchange removes a "man in the middle" that we have in our current process and drastically decreases the likelihood of unintentional human error.

The MCLOG M&S staff found that the RPA-based sim-IS architecture would decrease manpower requirements for their organization's sim-IS environments for staff training exercises (question 4).

> Used in conjunction with the batch file generator v2.0 this tool will easily decrease manpower requirements in excess of 60 Hrs per exercise. 2 personnel work an average of 6 hours to align data for 5 days.
>
> * * *
>
> …the [RPA-based sim-IS approach] would decrease the many person-hours for computing the log stats data and personnel readiness.
>
> * * *
>
> Yes, [the RPA-based sim-IS architecture would] significantly decrease manpower requirements.

The MCLOG M&S staff identified several operational information systems which they have considered including in sim-IS staff training environments but are infeasible to include based on existing sim-IS information exchange methods. They found that the RPA-based sim-IS architecture would allow their organization to introduce more information systems in staff training exercises beyond what they can currently support (question 5). Some of the specified information systems for future inclusion in staff training exercises included the Sea Service Deployment Module (SSDM), GCSS-MC, Storage Retrieval Automated Tracking Integrated System (STRATIS), and the Integrated Computerized Deployment System (ICODES) as well as personnel and casualty management systems.

> Yes. We have explored other systems such as SSDM, and GCSS and the requirements to load data have prevented their adoption. These systems do not have training variants that allow for the mass loading and management of data. An experiment to load data into GCSS-MC found it would take 8 months of work by 2 contractors to load 600,000 line items into the system. Timelines and requirements of this scope make its use [infeasible]. With RPA we would be able to identify the data and allow the system to self-populate the LOG AIS [logistics automated information system] over the course of time with little to no human requirements.

* * *

> Yes, it would allow staff to focus on other logistics training. Medical reporting, mortuary affairs, personnel S-1, Air C2, Naval C2 for integrated exercise.

* * *

> The RPA approach is definitely a method of incorporating other systems. In the future, this is a possibility. However, I don't see MCLOG integrating additional systems than what we currently manage for at least a year. We have a difficult time trying to formalize current processes and procedures with the systems we have now.

The MCLOG and BSC simulation staffs found that the prototype simulates timeliness/latency distributions and data entry error probabilities that would otherwise be too manpower intensive for the organization to achieve with conventional sim-IS information exchange methods (question 6).

> Absolutely…this process is currently replicated by the use of MSEL injects. The process developed here [with the RPA-based sim-IS architecture] would create a consistent environment that would force commanders and staffs to always validate information they utilize to make decisions. The excel reports would also allow the [exercise control] staff to track errors and the decision points made until their discovery.

* * *

> Yes. However, at this time, we do not purposely induce any latency or errors on the training audience during exercises [due to manpower requirements, training audience capabilities, and other limitations].

* * *

> Yes, because you’re taking out the human error and time dedicated per day.

* * *

> Yes, it will provide the latency required to validate the unit’s process.

During the demonstration, the MCLOG and BSC M&S staffs were shown the three RPA-generated reports (RPA schedule, supply status file, and entity status file) which can be used by the exercise control staff to manage the RPA workflows during execution. These reports are intended to support oversight of the RPA-based sim-IS information exchanges and to inform the exercise collections plan, informing OTCs of scheduled and executed STS dynamics (e.g., latency, late or early LogStat report submissions, or errors in LogStat reports), so they know what to look forward with regard to training audience reactions. The MCLOG and BSC simulation staffs found that these reports meet their organizations’ requirements for informing exercise control personnel on the status of scheduled/completed sim-IS information exchanges (question 7).

> Yes. Some understanding of the reports is required, but this is minimal. Within seconds of explaining the reports, a user can follow the logic to understand what errors have been introduced into the exercise.

* * *

> Yes, we can monitor the reports throughout the exercise, which will reduce person-hours [in] daily reconciliation.

c. Initialization of RPA, Simulation, and IS for RPA-based Sim-IS Environment

The final part of the MCLOG demonstration and interviews addressed the StartEx data generation and sim-IS synchronization process associated with the RPA-based sim-IS architecture. The process for StartEx data generation and synchronization of simulation and IS parametric databases for RPA-based sim-IS environments was described in Chapter IV. What differentiates the StartEx data generation and synchronization process for RPA-based sim-IS environments from the existing MCLOG StartEx data generation process for sim-IS environments is the requirement for explicitly aligning the nomenclature and unique identifiers used between the simulation and IS parametric databases.

Under the existing MTWS-CLC2S StartEx data generation approach, the BFG application is used to generate aligned MTWS and CLC2S StartEx files using a standard excel workbook file as the input for the BFG. This approach does not require identification of how entity and supply classes (and entity unique identifiers) manifest in simulation and IS user interfaces, as the pucksters who support manual sim-IS information exchange are assumed to be able to associate simulation supplies and entities with the corresponding supplies and entities within the appropriate IS, though with some effort. With the RPA-based sim-IS architecture, however, the sim-IS environment developers must address how supply classes, entity classes, and entity unique identifiers manifest in simulation and IS user interfaces, to ensure the RPA workflows are able to automatically associate supply classes and entity instances between a simulation and IS pair.

After being guided through the StartEx data generation and synchronization process for RPA-based sim-IS environments with the researcher, the MCLOG M&S section personnel were asked to compare the approach to their existing sim-IS initialization processes and identify any notable obstacles to its implementation in their organization (questions 12 and 13). The MCLOG M&S personnel found the proposed sim-IS StartEx data generation and synchronization process as less complex and less manpower intensive than their existing processes. This was partly attributed to the use of JTDS as the single system for management of scenario data, with all changes to unit structure, entities, and supply allocations managed within JTDS. With the BFG supporting CLC2S feed file generation based on the JTDS-generated OBS file as its input file, any last-minute exercise changes can be implemented in JTDS and synchronized across the simulation and IS feed files.

> The initialization and use of the [RPA] prototype is quite simpler, less manpower-driven, and much more reliable than our current methods. However, developing a deep understanding of the backend code to develop and modify the RPA prototype presents a steep learning curve. However, with enough time and practice, this can be overcome. I do believe the RPA prototype is worth pursuit and should become our primary method of "linking" JCATS and CLC2S [and MTWS and CLC2S].

* * *

> [The StartEx data generation process for the proposed RPA-based sim-IS environments is] less complex and a time saver.

While the StartEx synchronization process associated with the RPA-based sim-IS information exchange approach was identified as being less complex and less manpower intensive, it was also acknowledged that the proposed approach required greater effort in advance of a particular exercise. This refers to the extensive work required by the staff to maintain synchronized simulation parametric databases and ISs relative to the JTDS database. Such use of JTDS with synchronized simulation databases may, however, benefit MCLOG exercises even under the sim-IS construct without RPA. In their most recent exercise, for example, the MCLOG M&S staff reported encountering StartEx synchronization obstacles which prevented the exercising of their newly designed JCATS-CLC2S information exchange approach. This is the sort of StartEx data generation and synchronization problem that JTDS was initially developed to mitigate(Bowers et al., 2011). If the entire exercise scenario were maintained in JTDS, last minute changes to the scenario in terms of unit structure, entities, and supply allocations may have been accommodated under the proposed StartEx data generation approach.

The last question posed to the MCLOG M&S section was to identify what obstacles exist for the design of sim-IS environments which include timeliness/latency distributions and errors automatically occurring with specified probabilities (question 14). Obstacles identified by the MCLOG M&S personnel ranged from exercise design challenges to ensuring compliance with information assurance policies.

> The biggest obstacle we have for this is in the exercise design and control of intended latency and errors, not the actual execution…The fact that the RPA prototype already presents latency and forced errors is half the battle.

* * *

> Automatic introduction would require a mechanism to track its inject into the environment [and the reports adequately address this.] Second, in the case of software, [are the information assurance] and information technology policies to allow the use of RPA or other automation solutions to inject and manage the data. [With regard to information exchange error] probabilities, there is no

> study or data that support [the simulated] probability of errors. While we know this to be the case, the exact amount of errors is based on subjective analyses.

Although UiPath is currently included in the Marine Corps' Software Center, making its installation and implementation relatively simple from an IS and IT policy perspective, there are several additional RPA platforms which are not currently available in the Software Center. Additional obstacles with regard to RPA software are the need for a platform independent RPA conceptual modeling approach and future studies to evaluate the effectiveness and efficiency of the proposed RPA-based sim-IS architecture with other RPA platforms.

The lack of existing studies quantifying STS dynamics for most JCSs poses an obstacle for the design of RPA-based sim-IS environments. To some extent, this is a "chicken or egg" problem, as study of STS dynamics requires an environment suitable for studying the target dynamics. While historical analysis and field training can support some insights into STS dynamics, such studies would also require a more controlled environment which supports stimulation of operational ISs and simulation of some STS dynamics in order to study others. Section B of this chapter addresses how the proposed RPA-based sim-IS environments may serve as staged worlds, supporting the study of some of the very STS dynamics of JCSs which they are intended to simulate.

d. Comparing RPA-based Sim-IS Information Exchange and Recently Fielded MTWS-CLC2S Interoperability Function

This section addresses comparison of the RPA-based MTWS-CLC2S information exchange approach with the MTWS-CLC2S information exchange supported by recently fielded MTWS and CLC2S engineering changes. In 2016, the MCLOG M&S section registered a request for a MTWS-CLC2S information exchange function via a system change request (number 17462) presented to the MTWS configuration control board (Cole Engineering Services, Inc., 2018). The system change request was approved by the 2016 MTWS configuration control board. An initial MTWS-CLC2S information exchange prototype function was finally delivered to MCLOG for testing in 2022. The protracted development time has been ascribed to both the coronavirus pandemic (which began in 2019) and competing priorities for platform changes and modernization efforts (J. Tygart, personal communication, December 3, 2020). The engineered MTWS-CLC2S information exchange function includes the manual transfer of an MTWS-generated XML file by a puckster to a CLC2S client for upload. This is referred to here as the "program office MTWS-CLC2S information exchange approach" for simplicity in the discussion.

The MCLOG M&S section staff were asked to compare the utility of the two approaches (questions 8 and 9). Strengths and limitations were identified for each approach. One benefit of the program office approach is that it does not require alignment of StartEx data across MTWS and CLC2S. As discussed in the previous section, this is a considerable difference, especially for units which may lack the staff and expertise for StartEx data generation and synchronization. When the MTWS-generated XML file is loaded into CLC2S, all entities and supplies associated with each unit in MTWS are mirrored in the associated units in CLC2S. The previous entity and supply statuses are overwritten. While this facilitates a low-overhead initialization of MTWS-CLC2S exercise scenarios, it also results in a couple notable limitations. The program office solution writes over all entity readiness and supply status data in the corresponding units. Unlike the RPA-based approach, all entity and supply statuses are updated, with no option for selecting which entities and supply classes to have updated and which to leave for the training audience or exercise control to update manually. Furthermore, when MTWS entity statuses are loaded in CLC2S with the MTWS-generated XML, unique identifiers associated with the equipment and personnel do not reflect real world unique identifiers. Personnel, for example, are not represented in the program office approach with notional names and EDIPIs, as they are when using the RPA-based sim-IS approach.

Another notable limitation of the program office approach, particularly when compared to the RPA-based sim-IS information exchange approach, is that it does not support the simulation of STS dynamics like information latency and data entry error. Instead, ground truth data from the simulation is entered into CLC2S, presenting an unrealistically accurate and timely depiction of unit logistics statuses and potentially resulting in negative training for training audience staff and commanders. Such a simulation of STS dynamics was not specified as a requirement in the original 2016 request for MTWS-CLC2S information exchange, so additional system changes would need to be made to accommodate simulation of such dynamics in the program office approach.

> The program office fielded solution is very limited. It offers no unique identifiers for personnel and equipment. It requires a swivel seat solution to transfer data in the form of an xml file between MTWS and CLC2S. The upload would delete data inputted by a user and override it with sim data. This creates the potential for user errors to be brushed away and corrected by the simulation with no effect to the user. Further, the false creation of perfect data creates negative training as cited in the JDLM/BCS3 white paper.
>
> * * *
>
> For MCLOG the optimal solution is the RPA process because of the level of fidelity our exercises require. Initial or home station training would benefit from the program office solution but…advanced level training requires more variety and quantities of data and the ability to track each independently.
>
> * * *
>
> …the RPA process is better than the recently fielded MTWS-CLC2S information exchange function because it does not require human interaction. Furthermore, the RPA process is automated and consistent. The fielded MTWS-CLC2S IS function is simply an export/import of an .xml file, which still requires a user to copy a file from MTWS onto an external hard drive and import that file into a Windows 10 machine's web browser.

3. Follow-on Validation Work

While the field demonstration and subject matter expert interviews support a degree of validation regarding performance of the proposed RPA-based sim-IS architecture in support of the intended use, future work is necessary. The capabilities and limitations of this architecture and associated RPA modules should be validated in simulation-supported live training exercises. This can include a testing the architecture in parallel with conventional sim-IS information exchange as well as leveraging the architecture as the primary means of sim-IS information exchange in support of the training events. The RPA-based sim-IS architecture should also be validated for different types of environments and sets of simulations and ISs.

There are numerous types of sim-IS environments which may be supported by an RPA-based sim-IS architecture, and numerous types of STS dynamics which may be simulated. Validation of the RPA-based sim-IS architecture should be conducted for different types of sim-IS environments and different types of STS dynamics to identify suitability with respect to unique circumstances and inform the design and development of future sim-IS environments in support of training and wargaming. Several factors determine whether conventional or RPA-based sim-IS information exchanges are appropriate for different sim-IS environments, and continuous validation efforts will be necessary for informing sim-IS environment designers in how to weigh those factors in support of unique sim-IS environment requirements. The next section identifies how RPA-based sim-IS environments can function as staged worlds to support an iterative identification and refinement of the very STS processes and information propagation dynamics that they are intended to simulate.

B. SUPPORTING CONTINUOUS VALIDATION THROUGH USE OF SIM-IS ENVIRONMENTS AS STAGED WORLDS

> The effectiveness of a staged world rests in how effectively the essential properties of the cognitive systems triad are preserved in the experiences created – experiences that emerge from the relationship between people, technology, and work. A staged world can create situations that might arise only very seldom in naturalistic observation, while still preserving key properties of the work domain that create an authentic, immersive experience for practitioners. As a result, a staged world is an effective and efficient means of investigating cognitive work.(Smith & Hoffman, 2018, p. 108)

Validation of simulations is never truly complete if the referent or the intended use remains in flux. This is certainly the case for the simulation of JCSs and associated information propagation dynamics. For the proposed RPA-based sim-IS architecture, both the referent (STSs and associated information propagation dynamics) and intended use context (simulations and ISs) can be expected to undergo continuous changes as both JCSs and simulation capabilities evolve over time. For military and emergency response organizations, where the target JCSs are infrequently exercised in a real-world context, RPA-based sim-IS environments may provide a solution for this validation challenge. By leveraging RPA-based sim-IS environments as staged worlds for investigation of JCSs and associated information propagation dynamics (e.g., organizational process misalignment, information exchange latency, human data entry error), training organizations can continuously refine the representation of STS dynamics in their RPA modules.

In a now infamous quote, former Secretary of Defense Donald Rumsfeld once said "you go to war with the Army you have, not the Army you might want or wish you had at a later time" (Schmitt, 2004, p. 1). This statement provides a useful perspective for the issue addressed here. Unit staffs go through training at all levels of readiness, whether they are recently formed, conducting final preparations for a deployment, or somewhere in between. Similarly, the emergency operations center exercises conducted by local, state, or federal agencies reflect the people, processes, and technology they have at that time. Such exercises present training organizations an opportunity for documenting the processes and associated STS dynamics (e.g., error rates, latency) to inform the design of RPA modules. This can facilitate RPA-based sim-IS environments with high levels of validity compared to environments simply simulating doctrine and the levels of readiness that organizations "might want or wish [they] had at a later time."

There is precedent for the use of instrumented military training data to inform combat models (Rowland, 2019). Improvements in the instrumentation of live training in the field has increased the potential for reinvigorating the "cycle of research," where instrumented training data is used to inform wargames and analytic combat models (Perla, 2011; Perla et al., 2019). RPA-based sim-IS environments present an opportunity for a parallel *cycle of training*, where instrumented training environments inform the design of not just the three elements of JCSs, but also the training environments themselves. Of course, the process of leveraging RPA-based sim-IS environments to inform the simulation of sim-IS information exchange is more complicated than simply updating RPA modules to reflect the latest units' readiness. As with the cycle of research, the use of RPA-based sim-IS environments as staged worlds for investigation of cognitive work would also require historical analysis and additional field training environments to address the numerous STS dynamics to be simulated. Chapter VIII provides an overview of the state of the cycle of research and presents the concept for a cycle of training, where RPA-based sim-IS environments support the coevolution of JCSs and their associated training environments.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. DESIGN AND DEVELOPMENT OF SIM-IS ENVIRONMENTS: A SIM-IS ENVIRONMENT DSEEP OVERLAY

This chapter provides recommendations for the creation of a DSEEP overlay for RPA-based sim-IS environments. The recommendations are intended for a DSEEP overlay, rather than modifications to IEEE Standard 1730–2010 itself, as sim-IS environments expand the scope beyond the distributed simulation environments which DSEEP supports. Recommendations are presented in the context of both the DSEEP steps and the broader issues or themes which must be addressed across multiple DSEEP steps. Following a general discussion of these themes, more specific discussion of DSEEP recommendations are presented per step. This discussion is supplemented by the DSEEP overlay recommendations presented in Appendix H.

The recommendations provided here for a sim-IS DSEEP overlay are intended to support the design and development of sim-IS environments to simulate integrated business processes and associated STS dynamics. This is a primary difference between this recommended sim-IS DSEEP overlay and the C2SIM DSEEP overlay recommended by Heffner et al. (2014), which focused more on addressing syntax and semantics considerations of sim-C2 integration, as discussed in Chapter II. Another difference is the emphasis on supporting the design and development of RPA-based sim-IS information exchange mechanisms. The recommendations provided here for a sim-IS DSEEP overlay should be considered as complementary to the C2SIM DSEEP overlay considerations. Future work is needed to explore how the recommendations provided across these two proposed DSEEP overlays may be integrated to support the design of sim-IS environments that leverage the most efficient and effective sim-IS integration capabilities for achieving the desired sim-IS environments. Whether the selected means of sim-IS integration be conventional protocols, pucksters, C2SIM federations, RPA technology, or a combination of these mechanisms, the design of sim-IS environment should take the broader perspective presented here for ensuring the simulation of integrated business processes and STS dynamics.

A. THEMES FOR DSEEP WITH RPA-BASED SIM-IS ENVIRONMENTS

DSEEP considerations for RPA-based sim-IS environments can be roughly divided between those that address the simulation of sociotechnical systems and those that address the automation of sim-IS information exchange. Primary themes for RPA-based sim-IS environment issues requiring additional attention through a DSEEP overlay include:

Simulation of real-world sociotechnical systems and associated dynamics:

- Designing sim-IS environments to support training requirements
- Specification / alignment of various conceptual models
- Specification / simulation of target sociotechnical system dynamics

Automating sim-IS information exchange:

- Selecting sim-IS information exchange mechanism(s)
- Designing RPA modules to simulate BP-IS integration
- Sim-IS StartEx data generation and synchronization
- Mapping simulation and IS information (e.g., object classes and instances, status categories)
- RPA module documentation and reuse

These themes are provided to support discussion of obstacles in the design of sim-IS environments. They are not intended to suggest that the issues identified are separate or comprehensive. On the contrary, there are numerous interdependencies between issue themes, which are discussed to a limited extent throughout the chapter. The challenge of mapping simulation and IS information, for example, depends on the scoping of requirements conducted during modeling of sim-IS environments relative to the target BP-IS conceptual model(s) and is affected by the sim-IS information exchange mechanism selected for the sim-IS environment. These themes vary in their degree of RPA-specific considerations. Many are applicable for the design and development of any sim-IS environment, regardless of the sim-IS information exchange mechanism that is selected. A brief description of each theme is presented here before recommendations are provided for each DSEEP step.

1. Designing Sim-IS Environments to Support Training Requirements

The sim-IS environment design issue addressed here does not refer to step 3 of DSEEP. It refers to the question of how to scope the information representation for each of the envisioned sim-IS environment participant perspectives to support training requirements, as well as requirements for wargaming and/or analysis of JCSs. Sim-IS environments may support the development of participants' mental models and evaluation of JCSs through their simulation of how information is presented in context to decision-makers and staffs through context-appropriate information presentation mechanisms (e.g., digital ISs, operations center analog boards). They can also provide analysts insights into the capabilities and limitations of JCSs by exercising those JCSs in controlled environments. Design of sim-IS environments for training requires identification of the existing mental models and the target mental models for each of the sim-IS environment participant perspectives to be supported. Design of sim-IS environments for training, wargaming, and/or analysis requires identification of relevant environmental cues which are used by practitioners, from novice to expert, to support sensemaking and other macrocognitive functions (and/or microcognitive functions) in context.

In the context of DSEEP, this begins in Steps 1 and 2, with sim-IS environment objectives specification and conceptual analysis. The sim-IS environment designer must work with the sponsor to identify the requisite sim-IS environment participant perspectives and environmental cues to support requisite microcognitive and/or macrocognitive functions. The processes through which said cues are generated in the real world must also be modeled, in the form of BP-IS CMs and associated sim-IS environment CMs, to ensure the sim-IS environment presents participants' requisite information with adequate levels of fidelity in form and function. The resulting sim-IS environment requirements, BP-IS CM(s), and sim-IS environment CM(s) provide guidance for the rest of the DSEEP steps. This guidance will support the engineering and execution of sim-IS environments which simulate not only the objects and events of the target environment but also the way information about those objects and events are received by different participants. The sim-IS environment requirements and CMs also must be continuously evaluated over time as the elements of the target JCSs coevolve and the associated BP-IS CMs and target mental models themselves change.

2. Development, Maintenance, and Alignment of BP-IS, IS GUI, Sim-IS Environment, and RPA Conceptual Models

The design of sim-IS environments to support the different participant perspectives discussed previously requires identification of the information which must be presented for each perspective, the medium through which it should be presented, and how the information is presented in the context of the associated BP-IS integration. This information must be captured in the sim-IS environment conceptual model. Sim-IS environment conceptual models should be informed by BP-IS conceptual models which capture the designers' and sponsors' understanding of the integrated business process(es) to be simulated. While sim-IS environment conceptual models should be platform-independent in terms of the simulations and sim-IS information exchange mechanisms to be employed, they must be platform-specific for the operational ISs to be simulated. This includes details about each IS's graphics user interface, detailed BP-IS processes and procedures for how ISs are populated with requisite information, and information semantics and syntax requirements associated with each IS.

Where RPA is selected to support sim-IS information exchange, sim-IS environment designers must leverage at least four types of conceptual models to inform the selection and/or design of RPA modules and the overall sim-IS environment: (1) BP-IS conceptual models, (2) IS GUI conceptual models, (3) sim-IS environment conceptual models, and (4) RPA module conceptual models. The conceptual model for the sim-IS environment must be informed by a conceptual model (or models) for the target integrated business process(es) to be simulated, including the requisite STS dynamics to be simulated. These sim-IS environment conceptual models are then augmented by IS GUI conceptual models to inform the design of RPA module conceptual models specifying how RPA modules will support simulation of integrated business processes and associated STS dynamics.

The sim-IS environment conceptual model must capture what information is presented for each sim-IS environment participant perspective, and how that information is acquired, modified, and entered into the appropriate operational information system(s) based on the BP-IS CM(s) of the existing or envisioned integrated business process(es). The sim-IS environment conceptual model must identify the source of the information, the syntax and semantics of the information exchanged, the way(s) it is delivered to the requisite destination, and what factors influence the way it is delivered. This should include identifying situations where a misalignment may exist between how information is presented across different operational ISs due to the different processes that inform them. The sim-IS environment CM must also capture how the information is expected to be modified as it traverses the sociotechnical system before it is entered. Such modifications can be due to propagation effects, network effects, sociotechnical system dynamics, or cyber-attacks. STS dynamics can include information latency, variability in data entry/ extraction time, and changes to the information itself (e.g., human error, sensor error, value conversion errors).

The sets of conceptual models described so far can be viewed as providing a deliberate approach for addressing how ground truth information turns into perceived truth information in the real-world (BP-IS conceptual models) and how a sim-IS environment simulates that transition of ground truth information to perceived truth information (sim-IS conceptual models). In this way, the design of sim-IS environment CMs can be considered to constitute a deliberate approach for addressing the issue of the "end-user's perception" of the sim-IS environment, an issue identified by multiple NATO MSGs as needing to be addressed in a C2SIM DSEEP overlay(Heffner et al., 2014; Simulation Interoperability, 2015; Standardisation for C2-Simulation Interoperation, 2015).

The "end-users' perception" issue includes considerations across the range of interoperability levels. At the lower, technical level is the issue of "information overload," where receiving systems are provided more traffic than they can process (Heffner et al., 2014, p. 13). The sim-IS conceptual model should include identification of how such technical limitations of operational ISs impact the propagation of information. At the higher level, the challenge of ensuring ground truth is appropriately converted into perceived truth for sim-IS environment participants should be resolved by the alignment of the sim-IS conceptual model with the conceptual models of the simulated integrated business processes and associated STS dynamics. The scope of the sim-IS conceptual modeling considerations presented in this proposed sim-IS DSEEP overlay exceeds the considerations addressed in reports addressing a C2SIM DSEEP overlay. C2SIM DSEEP overlay recommendations focused on ensuring sim-C2 information exchange adequately accounts for the technical capabilities and limitations of operational C2 systems, with no mention of organizational processes or associated STS dynamics which may be associated with the real-world generation and propagation of information.

RPA module CMs provide greater detail regarding how exactly information is extracted from simulations, modified, and entered into IS GUIs to simulate the dynamics captured in the sim-IS environment CM and operational IS GUI CM(s). RPA conceptual models should be platform-independent, with regard to RPA software, to ensure generalizability across different RPA platforms. This facilitates the transition of RPA workflows should an organization need to transition to a different RPA platform. RPA CMs must include platform-specific details, however, for the specific operational IS(s) and the user interactions which the RPA module stimulates. Conceptual models for the specific simulations selected as member applications in a sim-IS environment also are necessary, during the development step, for informing the design of RPA modules for the extraction of information from simulations in a way that adequately aligns with the sim-IS CM.

It can be generally stated that the BP-IS CM informs the sim-IS environment CM, and RPA module CMs are informed by operational IS GUI CMs, the sim-IS environment CM, and member applications' CMs. It must also be noted, however, that sim-IS environments may also influence BP-IS CMs. Continuous maintenance of CMs for integrated business processes, ISs, RPA modules, and sim-IS environments is required to ensure sim-IS environments (including RPA modules) remain adequately aligned with the latest understanding of the target integrated business processes and operational ISs. For example, maintaining BP-IS and sim-IS environment CMs would support identification of the misalignment of JDLM and Marine Corps maintenance and supply processes, as discussed in Chapter IV, Section B.2. Sim-IS environments present a unique opportunity for military and emergency management organizations to evaluate rarely exercised integrated business processes, guiding the redesign of both. As integrated business processes are tested and refined in sim-IS training environments, the exercises may inform the redesign of BP-IS CMs. Associated sim-IS CMs would then need to be adapted to ensure a proper representation of the target integrated business process(es). This coevolution of BP-IS models and associated sim-IS conceptual models, illustrated in Figure 55, would necessitate the redesign of RPA-based workflows for sim-IS information exchange.

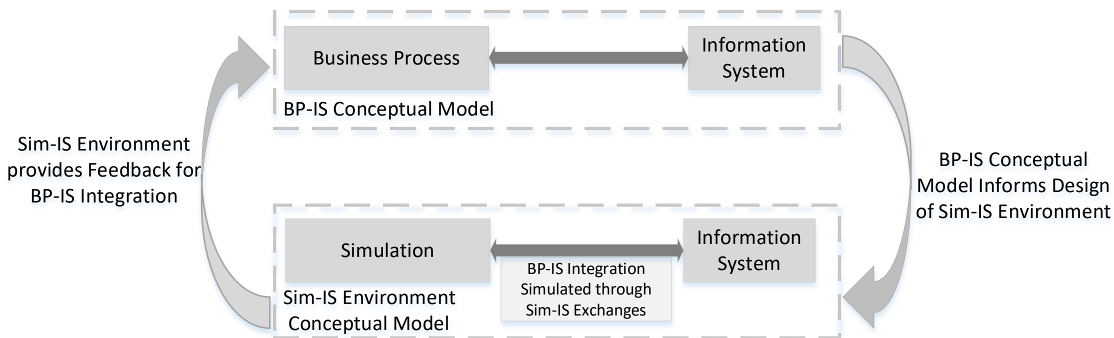


Figure 55. Coevolution of the BP-IS CM and Sim-IS Environment.

Sim-IS environments may be leveraged to exercise information systems and associated processes, to support training, or both. The use of training environments to inform analysis has been advocated by some who have highlighted the absence of human factors in wargaming and analysis as a notable limitation (Hanley, 2017; Rowland, 2019). The enhanced sim-IS environments proposed in this research present an opportunity for reinvigorating the role of training environments in the "cycle of research." These sim-IS environments also present an opportunity for establishing a parallel "cycle of training," where training environments coevolve with JCSs as the design of both the JCSs and the training environments themselves are continuously informed by training and field observations.

3. Specification and Simulation of Target STS Dynamics

The specification and simulation of STS dynamics should be addressed concurrently with the specification and simulation of other information degradation dynamics, including propagation effects, network effects, and cyber-attacks. The relationship between these effects was presented in Chapter II. While RPA-based sim-IS environments may be better suited for simulating STS dynamics, network effects, propagation effects, and some cyber-attacks may be better simulated through communications effects servers.

Selecting the information exchange dynamics to represent in a sim-IS environment, and how they are represented, should be informed by sponsor requirements specification and conceptual analysis. Cognitive task analysis methods conducted with naturalistic decision-making experts may support identification of key cues and dynamics for the target decision-making environment. Unfortunately for the DOD, some research suggests true naturalistic decision-making experts may be rare in the DOD due to servicemembers' limited time spent in each position (Shattuck et al., 2002). Furthermore, while CTAs can help identify the underlying structure of domains and key cues, additional methods may be required to evaluate the importance of different STS dynamics on decision-making. The real-time assessment model was developed to support analysis regarding the value of investments for decreasing information latency (Cundius & Alt, 2013). This model may support evaluation of processes to determine whether the effects of information latency on decision-making are significant enough to warrant its representation in a given sim-IS environment.

In addition to identifying what STS dynamics to include, the representation of STS dynamics in sim-IS environments can also require balancing the need to represent issues against a need to represent the frequency of their occurrence. This can necessitate simulating events with greater frequency than would be expected in the real world. One reason practitioners with more years of experience tend to have greater expertise, in addition to more finely tuned mental models, is their exposure to rare events. Developing that sort of expertise in a shorter time requires trainees to be exposed to different events with greater frequency than they might manifest in the real world. Determining how frequently to represent different STS dynamics in a sim-IS environment must include not just measurement of the real-world, but also training design considerations.

This issue of identifying how to represent STS dynamics in sim-IS environments is also directly related to the need for developing approaches for observation and instrumentation of sim-IS environments as well as field observation and instrumentation of operational ISs in field training and deployed environments. While sim-IS environments with STS dynamics informed by anecdotal evidence may provide a useful starting point for the iterative development of an understanding of true STS dynamics, field training, observation of deployed environments, and historical analysis should be sought out to improve the validity of the sim-IS environments.

4. Designing RPA Modules to Simulate BP-IS Integration

RPA workflows used in RPA-based sim-IS environments are unique from RPA workflows used in other work environments. A conventional approach to such an RPA workflow would include identification of the most efficient method for automating the exchange of requisite information, regardless of the process commonly employed by human operators. These RPA workflows serve to accomplish a task and free up human capital for other work. While the envisioned RPA workflows for sim-IS environments can free up human capital, they have an additional objective: simulating the actions taken by human operators (and the broader sociotechnical systems) to populate operational information systems in the simulated environment. This difference has important implications for the design of RPA workflows for sim-IS environments.

For example, consider the task of populating CLC2S with information about unit supply statuses derived from a combat simulation in a sim-IS environment. Human operators update unit supply statuses one unit at a time, with different operators updating statuses for their respective units. The designer of a conventional RPA workflow may find that the most efficient process for updating unit statuses in CLC2S includes the creation of a single excel file (i.e., feed file) which may be loaded into CLC2S to update the statuses for all units at once. This approach would likely be quicker and include fewer interactions with the CLC2S interface, facilitating the exchange of more information for more units than simulating the more common processes employed by CLC2S users navigating through the CLC2S interface to update supply statuses. While this is a more efficient process, it would not meet the requirement of simulating the dynamics of the sociotechnical process through which CLC2S is populated during real world operations.

RPA workflows built for sim-IS environments support more than the exchange of information between simulations and information systems; they simulate sociotechnical system dynamics. As simulations, the design of RPA workflows must be considered in the DSEEP standard to ensure the RPA workflows and the broader sim-IS environment are appropriately designed and executed to meet the requirements for the sim-IS environment. This chapter identifies unique considerations for the design and development of RPA-based sim-IS environments and lays the groundwork for a sim-IS DSEEP overlay.

5. Selecting Sim-IS Information Exchange Mechanism(s)

No one sim-IS information exchange mechanism is appropriate for supporting all information exchange requirements. The sim-IS information exchange mechanisms used in a sim-IS environment should be selected for each piece of information with consideration of multiple factors, including:

- The frequency with which sim-IS information exchange is required
- The speed with which sim-IS information exchange is required
- Suitability of standardized protocols for representing the information (e.g., unit locations with OTH-Gold)
- The information exchange capabilities of simulations and ISs to be used in the sim-IS environment (e.g., HLA compliant, capable of receiving OTH-Gold)
- STS dynamics which must be simulated (e.g., temporal or content quality degradation)
- Accuracy and precision required in sim-IS information exchange

- Ability of the process(es) for information extraction (from simulation(s)) and/or entry (into IS(s)) to be formally defined

Information exchange mechanisms provide different capabilities and limitations relative to these factors. Protocol-based approaches can support higher frequencies and higher speeds of sim-IS information exchange than RPA-based or puckster-based exchanges. When coupled with communications effects servers, they can support simulation of some information degradation due to network or propagation effects. Pucksters provide flexibility to support information exchange where adjudication is required (e.g., checking simulation values like casualty numbers before entering them into ISs) and where the process for information exchange and adjudication is difficult to formally define. Puckster-based approaches, however, can be expected to be limited in their ability to support high speed, high frequency, high accuracy, and high precision requirements. RPA-based approaches can support formally defined information exchange processes through user interfaces and simulation of STS dynamics. They are limited in their ability to support high speed and high frequency information exchange requirements, though these limitations can be mitigated by increasing the number of RPA bot instances for a sim-IS environment.

Sim-IS environment designers must consider these factors when selecting sim-IS information exchange mechanisms. The timely and efficient consideration of these factors depends on the availability of requisite information regarding the information exchange capabilities and interfaces of the simulation(s) and IS(s) to be used in a sim-IS environment. This should include documentation for RPA modules which may have been developed for the simulation(s) and IS(s), including conceptual models and documentation of performance characteristics like RPA module speed.

6. Sim-IS StartEx Data Generation and Synchronization

The process used for generation and synchronization of sim-IS environment StartEx data is important for multiple reasons. In addition to ensuring all member applications of a sim-IS environment are aligned to the degree desired at the start of a sim-IS environment, a thorough process is necessary to ensure sim-IS information exchange will be effective during sim-IS environment execution. The process must also be efficient to support the last-minute changes which often occur in the days (or hours) prior to the start of simulation-supported exercises. These can include changes to unit structures, force laydowns, and/or entity and supply allocations. If the generation and synchronization of sim-IS environment StartEx data is a manual process, such last-minute changes introduce great risk, as StartEx data adjustments will be prone to human error and the sim-IS environment designers may have little to no time to test the changes prior to execution. While the JTDS provides a mechanism for automating the generation and synchronization of StartEx data for distributed simulations in the form of OBS files, it does not automate the generation of StartEx data for operational ISs to be included in sim-IS environments. Additional guidance must be provided for sim-IS environments to ensure efficient, effective processes are established for the generation and synchronization of sim-IS environment StartEx data.

The importance of addressing initialization challenges unique to simulations and C2 systems in DSEEP was also highlighted by NATO MSG-085 in their proposed C2SIM DSEEP overlay. Heffner et al. (2014) identified system initialization as one of four major issues needing to be addressed by a C2SIM DSEEP overlay, along with including C2 stakeholders, time management, and the issue of "end-users' perception" addressed earlier. A potential tool they identify for addressing the initialization of C2SIM environments is the Scenario Initialization and Execution (SINEX) model, which would leverage initialization and data interaction specifications to support C2SIM member applications' initialization. While this approach may be beneficial for supporting sim-IS StartEx data generation and synchronization to some degree, RPA-based sim-IS information exchange necessitates an additional level of synchronization across sim-IS environment member applications owing to the nature of how RPA uses system GUIs.

For RPA-based sim-IS environments, sim-IS environment designers must ensure the way information manifests in simulation and IS user interfaces is aligned with the appropriate RPA data transformation components. The guidance for sim-IS environment StartEx data generation should address not only the general alignment of simulation and IS parametric data, but also the processes for ensuring simulation and IS information are adequately aligned with the data transformation components for associated RPA modules. The challenge of generation and synchronization of simulation and IS StartEx files is also directly related to the issue of mapping simulation and IS information. This includes the alignment of parametric data (e.g., DIS enumerations, nomenclatures) and the mapping of different information which may need to be represented across multiple member applications, as discussed in the next section.

7. Mapping Simulation and IS Information

While StartEx data generation and synchronization refer to the specification of synchronized scenario information across all simulations and ISs for a sim-IS environment, the mapping of simulation and IS information refers to the identification of how all requisite simulation information is represented in the requisite IS(s) information during the execution of a sim-IS environment. This includes the association of entity instances as they are represented in simulations and ISs as well as the association of simulated entity characteristics with the appropriate representation of those characteristics in the requisite IS(s). Guidance must be provided for addressing challenges associated with each of these.

When an entity-level simulation is employed in a sim-IS environment, the mapping of simulated entities to their equivalent representation in appropriate ISs can be relatively straight-forward. During the StartEx data generation and synchronization process, the relationship between simulated objects and IS representations for those entities should be established, providing a mapping. Even in these instances, however, the relationships are not always clear. Complications occur, for example, when the entities simulated by an entity-level simulation include "complex asset" entity classes. Complex assets can include a combination of equipment items which are tracked separately in the ISs. A complex asset representing a HMMWV "gun truck," for example, can include the vehicle, a crew-served weapon, and one or more radios. While the simulation represents all of these pieces of equipment as a single entity, the operational IS would likely represent each of these assets individually so that commanders and staff are aware of their available vehicles, crew-served weapons, and communications assets. The sim-IS designer must coordinate with the sponsor to determine how to represent the statuses of these three pieces of equipment, relative to the status of the simulated complex asset, to support the sponsor's requirements. Should the complex asset be damaged, for example, the sim-IS designer must identify how the damage is to be represented for the separate pieces of equipment associated with the complex asset.

An additional obstacle is encountered when aggregate simulations are employed in a sim-IS environment. The RPA-based sim-IS architecture presented in Chapter IV provides a way of resolving this issue, by maintaining a mapping of individual assets statuses in the ISs to the aggregate assets. The sim-IS environment designer must determine the appropriate way of adjusting these relationships over time. The default logic presented in the RPA-based sim-IS architecture is "first-in-first-out," where the first equipment to be damaged is also the first in the queue to be repaired. This is a simplification of how equipment and personnel statuses can be expected to rotate, as some personnel and equipment can be healed or repaired faster than others. The sim-IS environment designer must coordinate with the sponsor to determine the appropriate way of designing this aggregate-to-entity mapping to meet the sim-IS environment requirements.

Mapping simulation and IS information also includes the mapping of simulated object characteristics from simulations to their appropriate representation in ISs. Challenges here can be the result of limited fidelity in the simulation, the IS, or both. For equipment casualties, for example, entity statuses are often represented in simulations as mobility kill, firepower kill, mobility and firepower kill, or catastrophic kill. These damage categories do not clearly align with status categories used in some ISs to represent equipment damage (e.g., degraded or deadlined statuses shown in CLC2S). Furthermore, while one operational IS may only represent a piece of equipment as "deadlined" or "degraded," another may provide much more granular information regarding the type of equipment damage (e.g., reporting specific equipment defects in maintenance ISs like GCSS-MC). This can require sim-IS environment designers to map simulation entity characteristic categories to equivalent statuses in multiple ISs, even when those operational ISs are themselves misaligned in their representation of the ground truth situation. This issue is not limited to entity status representation. Another example is the reporting of unit supply statuses. When a simulation reports unit supplies with an aggregate number, as with fuel statuses discussed in Chapter I, the sim-IS designer may have to coordinate with the sponsor to determine how this information should be represented in the IS to represent what would be entered in real-world reporting, where the statuses of individual vehicle fuel tanks are unknown and unreported.

The design and development of sim-IS information exchange takes place across multiple DSEEP steps. It begins with the sim-IS environment designer coordinating with the sponsor to determine requirements for the representation of different information in the operational ISs. These requirements should inform the conceptual modeling of the target integrated business process(es) and the subsequent representation of these processes in the sim-IS environment conceptual model, and design of the sim-IS environment to facilitate representation of the information accordingly. While the conceptual modeling and design steps of DSEEP address the semantics and syntax considerations for how information should be presented in operational ISs and other mediums, syntax and semantics of sim-IS information exchange are addressed in the development step with the development of the simulation data exchange model.

8. RPA Module Documentation and Reuse

The conceptual modeling and documentation of RPA modules serves three related purposes: 1) ensuring RPA modules adequately simulate procedure(s) for entering information into ISs when necessary, 2) informing the design of sim-IS information exchange and the SDEM, and 3) facilitating RPA module reuse and providing performance characteristics of the RPA modules to inform decisions on their suitability for sim-IS environments. Ensuring RPA modules adequately simulate the dynamics specified in the sim-IS environment conceptual model depends on the alignment of RPA conceptual models with the requisite components of the sim-IS environment conceptual model. Facilitating the reuse of RPA modules across different sim-IS environments requires platform-independent RPA conceptual models and documentation of different RPA module performance characteristics which impact how it may support prospective sim-IS environments.

As with simulations and operational ISs, platform-independent conceptual models are necessary for informing the consideration of RPA modules as member applications and for determining how to allocate requisite functionality across member applications. RPA module conceptual models should identify the underlying structure and function of the RPA modules, this will aid sim-IS environment designers in determining the suitability of RPA modules for representing components of sim-IS environment conceptual model(s). Where the procedures employed for the entry of information into operational ISs impacts the presentation of information for different sim-IS environment participant perspectives, RPA conceptual models also must be aligned with conceptual models which capture the user interactions with IS GUIs to be simulated.

As discussed in Chapter II, the absence of an RPA domain ontology is an obstacle to the development of platform-independent RPA conceptual models. The absence of common RPA conceptual modeling practices and tools was reinforced by the results of a survey conducted with the Federal RPA Community of Practice (CoP) in support of this research. In January 2022, a survey was conducted to identify the extent of RPA conceptual modeling practices in the Federal RPA CoP and identify any common practices or tools. This survey, presented in Appendix G, was made available to all members of the Federal RPA CoP through the U.S. Government Services Administration's (GSA) Qualtrics survey tool. Prior to the dissemination of the survey, the survey questions were submitted to the Naval Postgraduate School IRB for a human subjects research determination review. The IRB found that the survey questions do not constitute human subjects research and therefore had no requirement for IRB approval.

Representatives of nine organizations provided responses for the survey, including Army Futures Command Headquarters, the National Institute of Food and Agriculture (NIFA), and the National Institute of Standards and Technology (NIST). Six of these organizations employ RPA for back-office process automation, two reported automating correspondence (e.g., emails), one reported automated development of reports, and three are reportedly exploring potential uses. Reported years of experience in employing RPA for the organizations range from zero to three years. Responses to questions about integrated business process modeling and RPA conceptual modeling approaches illustrated varied levels of rigor and equally varied modeling notations (Tables 16 and 17, respectively). Notations reported as being employed for integrated business process modeling included UML and a simplified version of BPMN coupled with “UX/UI mockups all in Excel.” Only one organization identified a standardized modeling language as being employed for RPA conceptual modeling: UML. Four of the nine respondents reported being unaware of any RPA conceptual modeling practices within their organization.

Table 16. Survey Responses Regarding Integrated Business Process Modeling.

Reported approach for integrated business process documentation in support of RPA workflow development	# of Organizations
No prior documentation of processes	2
Flow diagrams are received from users in assorted notations	1
High-level, standardized BP-IS conceptual models	3
Detailed BP-IS CMs documenting UIs and data formatting details	2

Table 17. Survey Responses Regarding RPA Workflow Modeling.

Reported approach for developing and maintaining RPA CMs	# of Organizations
The only documentation is that provided by the RPA platform	2
High-level RPA CMs are maintained	1
RPA CMs provide details of UI interactions and data formatting	2

When asked about their organizations’ processes for updating RPA CMs and CMs for integrated business processes, similarly varied responses were observed, as seen in Table 18. The responses to this question illustrated the limitations of using a survey for collecting insights about different organizational conceptual modeling approaches. Several of the responses received suggest that the respondents misunderstood the question or did not appreciate the intended emphasis on understanding whether organizations address the alignment of RPA and integrated business process CMs and how.

Table 18. Survey Responses Regarding Business Process and RPA Modules CM Alignment.

Reported organization processes for updating RPA CMs and CMs for associated integrated business processes as they change	# of Organizations
No, we don't maintain RPA and associated business process CMs	3
No, we only maintain RPA and business process CMs through development	2
Yes, we maintain and update models of RPA workflows and associated business processes over time	3

Although three of the nine organizations responded that they employ an RPA conceptual modeling notation that is platform-independent, one of the three organizations reported earlier in the survey that the only tool they use for RPA conceptual modeling is the RPA workflow documentation provided by the RPA development platform they use. As these responses are not compatible, this illustrates another example of the limitations of survey for questions about organizational conceptual modeling practices. Despite attempts to frame the discussion by providing a description of conceptual modeling at the beginning of the survey (as seen in Appendix G), the survey proved an inadequate means of collecting information about different organizations' processes. Future attempts to determine the state of RPA CM practices may be better served by employing interviews which afford an opportunity to ensure respondents' understanding of the questions and secure a more thorough understanding of their responses.

Despite the noted limitations of the survey as a means of gaining insight into the practices of different Federal RPA CoP members, several insights can be taken away from this survey. None of the organizations reported using a standard technical RPA workflow language for the automated translation of RPA workflows between different RPA platforms. As with industry, among the Federal RPA CoP survey participants, RPA is primarily employed in support of back-office process automation. The Federal RPA CoP does not maintain a set of best practices for the development and management of RPA conceptual models, for the development and management of integrated business process conceptual models, or for the continuous alignment of CMs for RPA workflows and the integrated business processes they are intended to support. Finally, the apparent miscommunication observed in the last few questions may be the result of a limited appreciation of the role of conceptual models in support of BP-IS alignment.

The documentation of RPA modules has implications beyond alignment with the sim-IS environment conceptual model. The speed of RPA workflows should be compared to the requirements for the number of pieces of information which need to be exchanged and the speed with which information exchanges occur. This problem is not unique to RPA-based sim-IS information exchange. It is one of the factors which should be considered when determining the appropriate sim-IS information exchange mechanism(s) for a sim-IS environment. RPA module performance documentation should be maintained in system design documentation along with RPA conceptual models to aid sim-IS environment designers in determining the appropriate sim-IS information exchange mechanism for each piece of information to be exchanged and the requirements associated with each, such as timeliness, frequency, and simulation of STS dynamics. In addition to guiding the initial design of RPA modules which adequately simulate target business processes, RPA module conceptual models should facilitate the continuous evaluation and alignment of the RPA module with the IS GUI(s) and associated process(es). ISs, their GUIs, and the associated work processes continuously change. As ISs change, RPA modules must be adapted to ensure they align with both the IS and associated processes for interacting with the IS GUIs. RPA module conceptual models should facilitate this analysis.

B. DSEEP OVERLAY RECOMMENDATIONS

Multiple DSEEP overlays have been published by IEEE to provide more specific guidance on the employment of DSEEP for specific contexts or issues. The DSEEP Multi-Architecture Overlay (DMAO) (*IEEE Std 1730*.1, 2013) provides guidance for the engineering and execution of distributed simulation environments where multiple distributed simulation architectures are involved, with an emphasis on the three most common simulation interoperability approaches: DIS, HLA, and Test and Training Enabling Architecture (TENA). While the DMAO provides some guidance for consideration of "secondary communications" and the intentional transmission of "non-ground-truth data" by member applications like communications effects server, this issue is addressed primarily to support non-ground truth data differentiation from ground-truth data by all requisite member applications. It does not guide the design and development of environments for the intentional simulation of information degradation. The DSEEP Verification and Validation Overlay (in IEEE balloting at time of this writing) is intended to "provide a more detailed view of the V&V process implied by the DSEEP" (*SISO*, 2022).

These overlays supplement the DSEEP with additional considerations for issues to be expected when implementing the DSEEP for their respective environment or purpose. While the structure of each overlay is different, their structures mirror and build upon that of the DSEEP standard. In the DMAO, for each DSEEP step, the associated activities are reviewed to determine whether overlay-specific considerations must be addressed. Any "issues" identified with each subordinate step are addressed with a description of the issue and recommended action(s) for its resolution. After identification of all issues for a given subordinate step, recommendations are presented for overlay-specific additions to the inputs, tasks, and outcomes for the DSEEP step. The VV&A Overlay navigates through each DSEEP step and subordinate activity, identifying VV&A considerations. Unlike the DMAO, which provides additional guidance to simulation environment designers/ developers for multi-architecture environments, the VV&A overlay provides guidance to VV&A personnel. This section uses the structure of the DMAO (*IEEE Std 1730*.1, 2013) as a template, providing an overview of DSEEP steps and only providing recommendations where existing DSEEP activities are insufficient for the engineering and execution of RPA-based sim-IS environments. A detailed representation of DSEEP is presented in Figure 56.

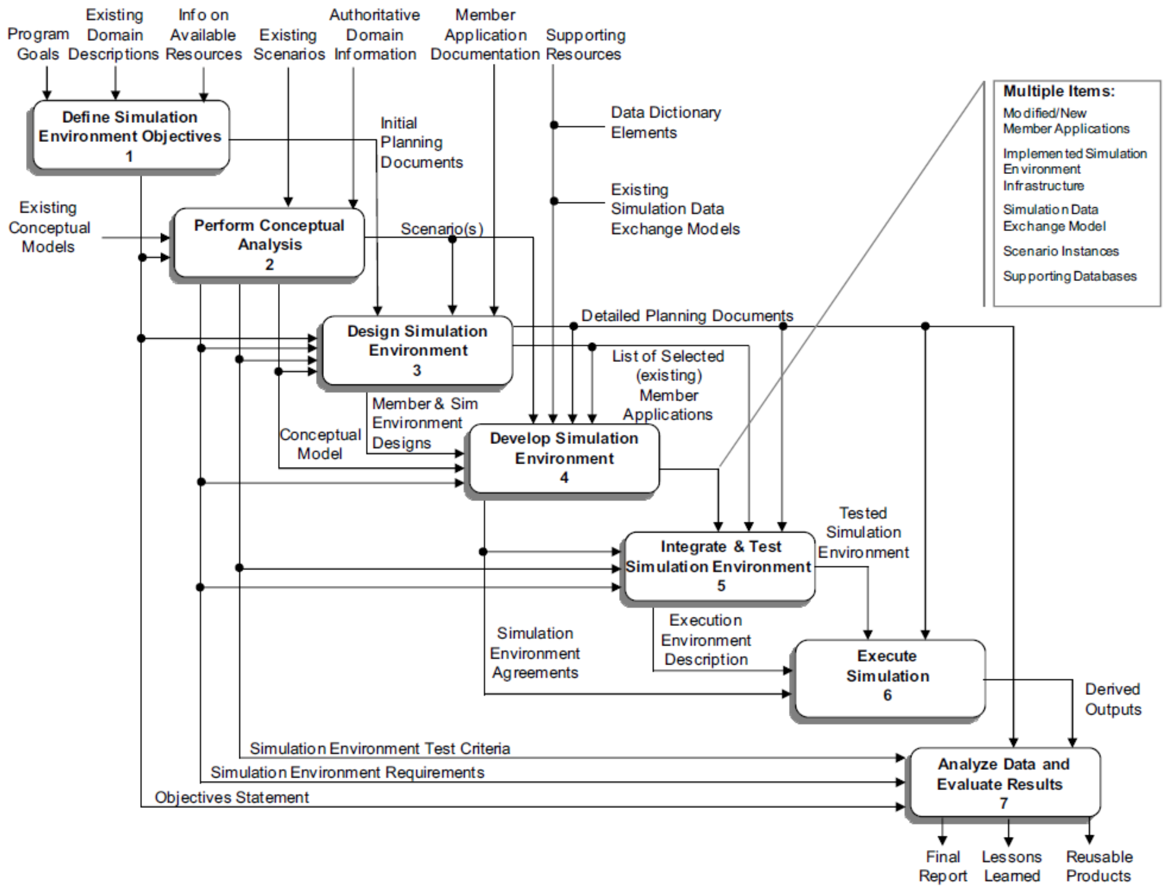


Figure 56. Detailed DSEEP View. Source: *IEEE Std 1730* (2011) © 2011 IEEE.

The inputs, tasks, and outputs of DSEEP are specified for each step. In this section, each step is reviewed, and recommendations are made regarding special considerations which must be addressed for RPA-based sim-IS environments. It is important to remember that DSEEP is not a one-way process. While the process progresses generally from step 1 to 7, DSEEP supports corrective actions and iterative development, as seen in the top-level DSEEP process flow presented in Chapter II. For example, simulation environment developers may only discover interoperability limitations which are unacceptable after reaching step 4 (Develop Simulation Environment) or step 5 (Integrate and Test Simulation Environment), requiring a return to step 3 (Design Simulation Environment) for reevaluation of potential member applications.

1. Step 1: Define Simulation Environment Objectives

> The purpose of Step 1 of the DSEEP is to define and document a set of needs that are to be addressed through the development and execution of a simulation environment and to transform these needs into a more detailed list of specific objectives for that environment. (IEEE Std 1730, 2011, p. 9)

Figure 57 illustrates the activities associated with Step 1 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

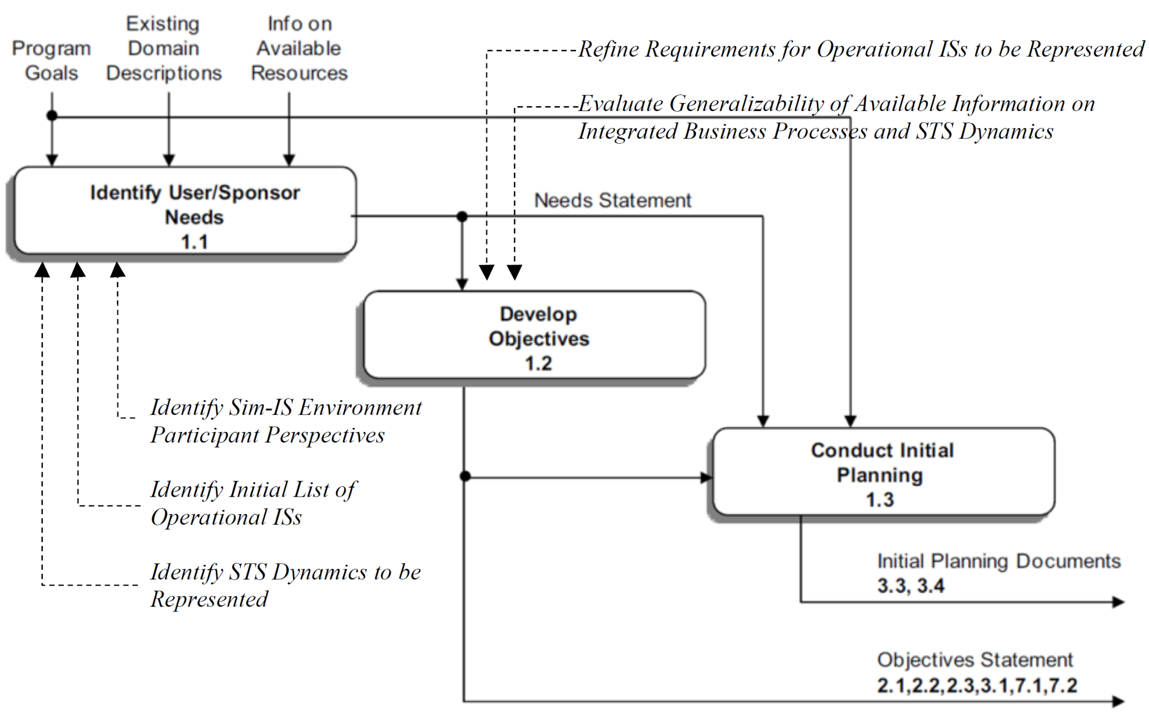


Figure 57. DSEEP Step 1: Define Simulation Environment Objectives. Source: IEEE Std 1730 (2011) © 2011 IEEE.

a. Activity 1.1: Identify User/Sponsor Needs

> The primary purpose of this activity is to develop a clear understanding of the problem to be addressed by the simulation environment. The needs statement may vary widely in terms of scope and degree of formalization. It should include, at a minimum, high-level descriptions of critical systems of interest, initial estimates of required fidelity and required behaviors for

> simulated entities, key events and environmental conditions that must be represented in the scenario, and output data requirements. In addition, the needs statement should indicate the resources that will be available to support the simulation environment (e.g., funding, personnel, tools, facilities) and any known constraints that may affect how the simulation environment is developed (e.g., required member applications, due dates, site requirements, security requirements). In general, the needs statement should include as much detail and specific information as is possible at this early stage of the DSEEP. (IEEE Std 1730, 2011, p. 9)

(1) Issue 1.1.1: Identify Sim-IS Environment Participant Perspectives

Description. Sim-IS environments can often be employed in various ways to support different requirements. A constructive simulation like MTWS can support staff training for companies, battalions, regiments, divisions, and higher command structures. The simulation serves a different purpose for different participants, however, and the design of the sim-IS environments requires an understanding of the participants' perspectives relative to the simulated environment, how they would receive information regarding that environment, and the STS dynamics associated with how that information would be provided to them. The participants in sim-IS environments (i.e., trainees if the simulation is used for training purposes) should be presented information about what is occurring within the simulation environment in accordance with their respective perspectives.

Recommended action(s). The intended use of the sim-IS environment should be identified early in the DSEEP. The intended use should include identification of the purpose of the sim-IS environment (e.g., support training, experimentation, analysis), the organizations to be supported, and the particular positions or perspectives of the intended sim-IS environment participants. This can also include identification of the particular processes or responsibilities to be exercised (e.g., casualty tracking, targeting, kill chain management). The sim-IS environment designer should work with the sponsor to talk through all possible sim-IS environment participants for the intended use and how they are intended to interact with the sim-IS environment. The sponsor may not be able to identify all the requisite perspectives and/or processes to be supported by the sim-IS environment, but they can identify which participants and processes must be supported. During Step 2 (conceptual analysis), the sim-IS environment designer can identify additional perspectives and/or processes relative to the intended use that may have been missed during this step. At that time, the sim-IS environment designer, in coordination with the sponsor, may return to this step to adjust the sponsor's requirements.

(2) Issue 1.1.2: Identify Initial List of Operational ISs for Sim-IS Environment

Description. Depending on the intended use of the sim-IS environment, the sponsor may know operational ISs that must be represented in the sim-IS environment and what functions or interfaces for those ISs must be stimulated to support the intended use. A sponsor's initial list of operational ISs can support identification of ISs to be explored or it can limit the scope of the sim-IS environment. If, for example, the sponsor requires a sim-IS environment to train battalion staffs to leverage all available ISs to manage equipment readiness, the sim-IS environment designers should consider representation of CLC2S, TCPT, and GCSS-MC in their sim-IS environment, exploring all possible ways they can be employed in the conceptual analysis step. If, however, the sponsor is only interested in training staffs to manage equipment readiness in an environment where only CLC2S is available, this limits the scope of the sim-IS environment and helps scope the conceptual analysis.

Recommended action(s). As with the identification of requisite sim-IS environment participant perspectives, the sponsor may not be initially aware of all ISs which should be represented in the sim-IS environment. An initial list of required operational ISs should be developed in coordination with the sponsor. Each IS should also be associated with the sim-IS environment participant(s) intended to leverage it, and for what purpose(s), to the extent possible.

(3) Issue 1.1.3: Identify STS Dynamics to be Represented

Description. DSEEP states that this activity should include identification of sponsor needs including "initial estimates of required fidelity" (p. 9). The appropriate level of fidelity for a particular sim-IS environment depends on the intended use of that sim-IS environment and, for training environments, the competence of the intended training audience. For introductory training on staff processes, it may not be necessary to simulate the exchange of information across the target STS with a high level of fidelity. A high-fidelity representation of information exchange dynamics like information latency, cyber-attacks, and radio propagation degradation may even overwhelm novice trainees, decreasing training effectiveness.

Recommended action(s). For sim-IS environments the question of sim-IS environment fidelity should explicitly address what STS information exchange dynamics should be represented. This should include identifying the categories of information exchange dynamics that are desired (e.g., STS dynamics, network effects, radio propagation effects, types of cyber-attacks) and the purpose of representing the dynamics. The purpose of representing the dynamics may be just to expose the sim-IS environment participants to different dynamics they may encounter, or the purpose may be to evaluate the suitability of their organizational processes and develop their intuition for a specific operating environment. The latter purpose may require a different level of fidelity for the distributions for different information exchange dynamics simulated in the sim-IS environment.

(4) Sim-IS Environment-Specific Inputs, Tasks, and Outcomes for Activity 1.1

Listed below are recommendations for additions to the inputs, tasks, and outcomes specified in the DSEEP for activity 1.1. This section provides an example of how this detail may be presented for a DSEEP overlay for RPA-based sim-IS environments, based on the DMAO as a template. Overlay recommendations for additional inputs, tasks, and outcomes for the remainder of the DSEEP steps and activities are specified in Appendix H.

- Sim-IS Environment-Specific Inputs
- - No additions to what is specified in DSEEP
- Sim-IS Environment-Specific Tasks

- Identify intended use for sim-IS environment with the sponsor, and the requisite sim-IS environment participant perspectives to be supported

- Identify operational ISs to be represented in sim-IS environment and how they are expected to be employed in support of each sim-IS environment participant perspective

- Identify the STS dynamics to be represented, the purpose for their representation, and the requisite levels of fidelity to support the purpose(s)

- Sim-IS Environment-Specific Outcomes

- Description of intended use of the sim-IS environment, with description of intended sim-IS environment participant perspectives to be supported

- Initial list of operation ISs to be represented in sim-IS environment, including function and form requirements for each

- Initial list of STS information exchange dynamics to be represented and requisite levels of fidelity

b. Activity 1.2: Develop Objectives

The purpose of this activity is to refine the needs statement into a more detailed set of specific objectives for the simulation environment. The objectives statement is intended as a foundation for generating explicit simulation requirements (i.e., translating high-level user/sponsor expectations into more concrete, measurable goals). This activity requires close collaboration between the user/sponsor of the simulation environment and the development/integration team to verify that the original needs statement is properly analyzed and interpreted correctly, and that the resulting objectives are consistent with the stated needs.

Early assessments of feasibility and risk should also be performed as part of this activity. In particular, certain objectives may not be achievable given practical constraints (such as cost, schedule, and availability of personnel or facilities) or even limitations on the state-of-the-art of needed technology. Early identification of such issues and consideration of these limitations and constraints in the objectives statement will set appropriate expectations for the development and execution effort. (IEEE Std 1730, 2011, p. 11)

(1) Issue 1.2.1: Refine Requirements for Operational ISs to be Represented

Description. Following identification of the initial list of operational ISs to be represented in the sim-IS environment, the form and function required for the representation of each IS must be identified. Refining what is required with regard to the form and function of the operational ISs may assist the sim-IS designer in determining whether an emulator or training server is adequate or if an operational system is necessary to meet the sponsor's requirements. These requirements must also identify the process(es) associated with the operational IS(s) and the information that must be presented to the operational IS(s) to simulate the target integrated business process(es).

Requirements for the information to be presented to operational IS(s) are necessary for guiding the design of sim-IS information exchange mechanisms in later DSEEP steps. Two types of issues to be considered are identifying requirements for the semantics of information presented to ISs and even requirements for simulating the misalignment of information presented across multiple ISs. An example of the former issue is mapping entity characteristics categories, such as identifying the appropriate relationship between different personnel casualty status options in an IS (e.g., ambulatory or litter and urgent, priority, or routine) and common simulation casualty statuses (e.g., mobility-kill, firepower-kill, mobility and fire-power-kill, catastrophic-kill). Such mapping could range from a precise one-to-one mapping of statuses (e.g., report all mobility-kill casualties as priority-litter casualties), or it could include a random selection of categories based on specified probabilities (e.g., report 80% of mobility-kill casualties as priority-litter and 20% as priority-ambulatory). This is an issue for simulation-to-simulation interoperability as well, as with the exchange of casualties between MTWS and JDLM, which affords a higher fidelity representation of casualty statuses.

Representing the misalignment of information presented in multiple operational ISs requires consideration of how the different organizational processes associated with different operational ISs may result in different information being presented in different operational ISs regarding what appears to be a common status. This issue was addressed in Chapter IV, Section A.5, where it was discussed in terms of intentionally simulating lapses in semantic and pragmatic interoperability of real-world ISs. For example, a common issue encountered with readiness reporting is for a logistics IS used by equipment operators to reflect a particular piece of equipment as being unavailable, while an operational maintenance IS shows it in an operational status. This can cause friction within an organization, as decision-makers attempt to determine what the real readiness is for the organization. The misalignment of the ISs can reflect a simple delay, as one IS was updated prior to the other IS, or it could reflect a more substantive misalignment between operations and maintenance processes.

Recommended action(s). In addition to identifying the form and function of the ISs to be represented, the nature of the information should be identified. This should include, for example, whether information is scheduled (e.g., logistics status reports, equipment readiness reports) or event-based (e.g., combat reports, information relating to commander's critical information requirements). Such a thorough understanding of the nature of the information being represented in the operational IS will inform not only what is necessary for representing the operational IS, but also identifying the sim-IS information exchange mechanism(s) best suited for simulating the target integrated business process(es). Requirements for the information presented should include consideration of the different types of information categories and any requirements for simulating misalignment of information presented in multiple operational ISs.

(2) Issue 1.2.2: Evaluate Generalizability of Information Sources for Integrated Business Process Design and STS Dynamics Estimates

Description. Although operational ISs may be available for use across an organization, that does not mean all components of the organization use the ISs in the same way. In some instances, integrated business processes may be clearly defined and standardized across the enterprise, ensuring all units or components of an organization employ an IS in a uniform way. Organizations may provide best practices to guide the design of unit-specific processes for leveraging ISs. Some ISs are also fielded without standardized processes or best practices to guide the implementation of the IS in support of business processes. This is the case for some Marine Corps logistics ISs, with units employing ISs like CLC2S and TCPT in different ways. Before proceeding with the design of sim-IS environments to simulate integrated business processes, sim-IS environment designers must determine the degree to which integrated business processes have been designed and standardized across the potential participants in the envisioned sim-IS environments. Where integrated business processes have not been standardized, the sim-IS environment may need to be designed to support process redesign for each scenario, just as task-organized units must redesign misaligned processes. This necessitates coordination with the sim-IS environment sponsor to determine the appropriate sources of information regarding integrated business processes to simulated.

Information regarding the nature of information degradation across a sociotechnical system can also be difficult to acquire. At times, expected STS dynamics may only be informed by anecdotal evidence or estimates by subject matter experts. Sim-IS designers may also have a limited understanding of how radio propagation effects, network effects, and how cyber-attacks may manifest for a particular scenario. In every situation where such STS information degradation effects are to be simulated, the sim-IS environment designer must evaluate the available information regarding the effects and coordinate with the sponsor to determine if representation of the effects with the available information is worth the risk of misrepresenting the dynamics.

Recommended action(s). The sim-IS designer should discuss the available sources of information regarding integrated business processes and associated STS dynamics with the sponsor to determine whether processes are adequately standardized for common simulation and identify sources considered reliable enough to inform the design of the processes and desired STS dynamics to be represented.

c. Activity 1.3: Conduct Initial Planning

> The purpose of this activity is to establish a preliminary simulation environment development and execution plan. The intent is to translate the objectives statement, along with the associated risk and feasibility assessments, into an initial plan with sufficient detail to effectively guide early design activities. The plan may effectively include multiple plans, and should cover such considerations as verification and validation (V&V), configuration management, and security. The plan should also address supporting tools for early DSEEP activities, based on factors such as availability, cost, applicability to the given application, ability to exchange data with other tools, and the personal preferences of the development/ integration team.
>
> The plan should also define a high-level schedule of key development and execution events, and provide additional scheduling detail for all pre-development (i.e., prior to Step 4) activities. Note that the initial plan will be updated and extended as appropriate in subsequent development phases as additional information is accumulated throughout the evolution of the distributed simulation design...(IEEE Std 1730, 2011, p. 12)

No additional considerations are required for extending this activity to support RPA-based sim-IS environments.

2. Step 2: Perform Conceptual Analysis

> The purpose of this step of the DSEEP is to develop an appropriate representation of the real-world domain that applies to the defined problem space and to develop the appropriate scenario. It is also in this step that the objectives for the simulation environment are transformed into a set of highly specific requirements that will be used during [sic] design, development, testing, execution, and evaluation. (IEEE Std 1730, 2011, p. 13)

Figure 58 illustrates the activities associated with Step 2 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

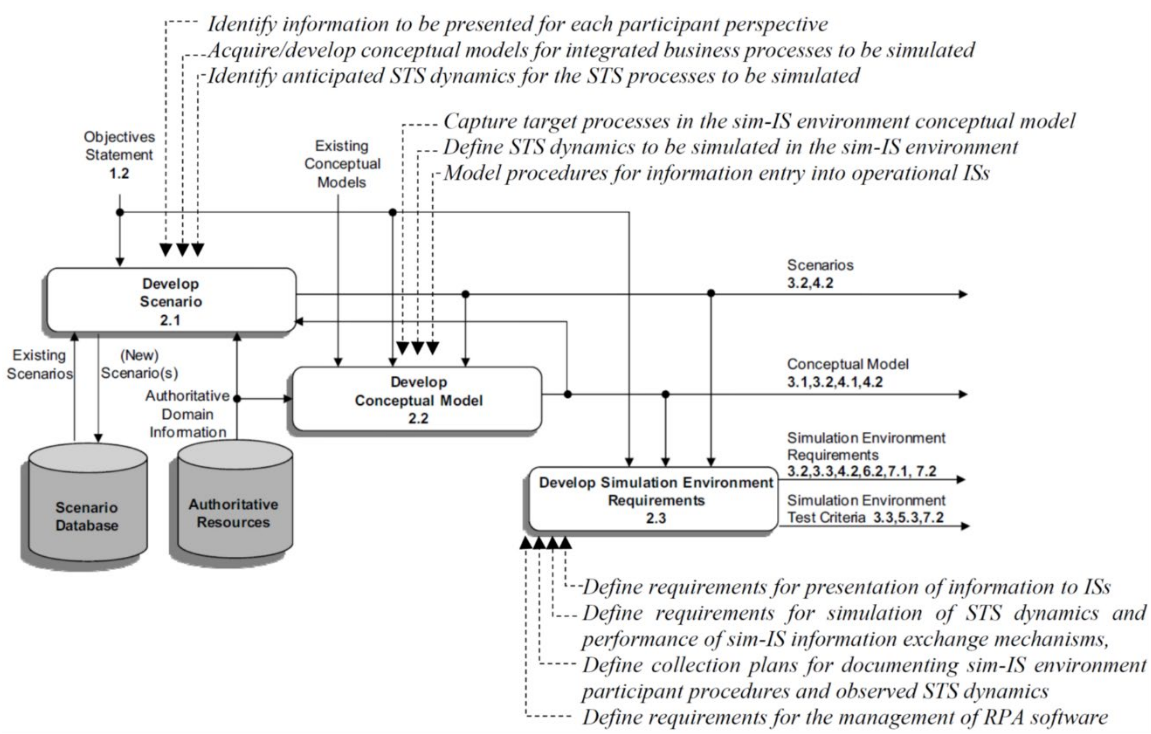


Figure 58. DSEEP Step 2: Perform Conceptual Analysis. Source: *IEEE Std 1730* (2011) © 2011 IEEE.

a. Activity 2.1: Develop Scenario

The purpose of this activity is to develop a functional specification of the scenario. Depending on the needs of the simulation environment, the scenario may actually include multiple scenarios, each consisting of one or more temporally ordered sets of events and behaviors (i.e., vignettes). The primary input to this activity is the domain constraints specified in the objectives statement (Step 1), although existing scenario databases may also provide a reusable starting point for scenario development. Any additional input is provided by the conceptual model, which may be developed in parallel with the scenario. Where appropriate, authoritative sources for descriptions of major entities and their capabilities, behavior, and relationships should be identified prior to scenario construction. A scenario includes the types and numbers of major entities that must be represented within the simulation environment, a functional description of the capabilities, behavior, and relationships between these major entities over time, and a specification of relevant environmental conditions (such as urban terrain versus natural area, type of terrain, day/night, climate, etc.) that impact or are impacted by entities in the simulation environment. Initial conditions (e.g., geographical positions for physical objects), termination conditions, and specific geographic regions should also be provided. The product of this activity is a scenario or set of scenarios, which provides a bounding mechanism for conceptual modeling activities.

The presentation style used during scenario construction is at the discretion of the simulation environment development/integration team. Textual scenario descriptions, event-trace diagrams, and graphical illustrations of geographical positions for physical objects and communication paths all represent effective means of conveying scenario information. Software tools that support scenario development can generally be configured to produce these presentation forms. Reuse of existing scenario databases may also facilitate the scenario development activity. (IEEE Std 1730, 2011, p. 14)

(1) Issue 2.1.1: Identify What Information Must Be Presented for Each Sim-IS Environment Participant Perspective and How

Description. Different scenarios have different requirements for information which must be presented to support sim-IS environment participants' decision-making and staff work. The first step in determining what operational ISs to represent and how to populate them is to determine what information must be presented to support each of the sim-IS environment participant perspectives. This should lead to the identification of appropriate mediums for conveying that information (e.g., analog COC boards or operational ISs).

Identifying the specific cognitive processes to be supported with said information, including sensemaking, decision-making, and the staff work that informs decision-making, is also important for ensuring the appropriate cues are included in the information and in its manner of presentation for the respective sim-IS environment participant perspectives. The operational ISs to be included in a sim-IS environment must be appropriate for the organization and its processes, but also for the scenario.

Recommended action(s). After identifying "the entities, behaviors, and events that need to be represented in the scenario(s)" (IEEE Std 1730, 2011, p. 14), the sim-IS environment designer should identify what information needs to be conveyed to the different sim-IS environment participants about said entities, behaviors and events. This also should include identification of the appropriate medium(s) (e.g., what operational ISs) for conveying which pieces of information to support each sim-IS environment participant perspective.

(2) Issue 2.1.2: Acquire or Develop Conceptual Models for STS Processes to be Simulated

Description. Following identification of what information must be presented to sim-IS environment participants, conceptual models must be acquired or developed for the real-world integrated business processes which populate the operational ISs with the requisite information. For sim-IS environments, the very existence of an operational IS which provides sim-IS environment participants insight into simulated events and/or characteristics of simulated entities means that some real-world process exists through which information is collected and entered into the operational IS. The process can include automated data exchange, with no human in the loop, as with GPS signals populating BFTs or other C2 systems. It can include only humans and no electronic systems, as with a runner manually updating information on a COC dry erase board. Often, the information presented to sim-IS environment participants is generated, transmitted, and presented in the real world through sociotechnical systems including both automated and human actions.

Before the conceptual model for a sim-IS environment can be developed, conceptual models must be acquired or developed to capture how information is, or at least is believed or intended to be, acquired in the real world, synthesized, and entered into the requisite operational ISs. As discussed in Chapter II, this can be quite challenging. Many organizations do not maintain conceptual models documenting their business processes and how ISs are integrated into those business processes.

Recommended action(s). Conceptual models should be acquired or developed for the real-world integrated business processes to be simulated through the presentation of information for the sim-IS environment participant perspectives. This should include identification of the source of the ground-truth information, the operational IS(s) through which each piece of information is presented to the respective role(s), and the processes through which each piece of information is obtained and delivered to the respective operational IS(s) in the real world. Ideally, such conceptual models for integrated business processes would be maintained by the respective organizations to support iterative improvements in the processes, personnel competencies, and associated ISs.

(3) Issue 2.1.3: Identify Anticipated STS Dynamics for the STS Processes being Simulated in the Sim-IS Environment

Description. Before developing the sim-IS environment conceptual model, the sim-IS environment designer must determine if the integrated business process conceptual models developed or acquired in Issue 2.1.2 are prescriptive or descriptive. If they are prescriptive, the sim-IS environment designer must determine the degree to which they are expected to be implemented as prescribed in the scenario. Are the personnel expected to understand the processes and execute them as prescribed? How might different organizations execute the processes differently? If they are descriptive models, the sim-IS designer must evaluate the source, in coordination with the sponsor, to determine whether the models adequately represent how the processes are expected to be executed in the scenario. The sim-IS environment designer must also identify anticipated STS dynamics (including network effects, propagation effects, etc.), similarly evaluating any sources of authoritative information to determine if they provide an acceptable representation of the anticipated degradation of information across the STS(s). This issue is similar to considering the simulation of entity movements and rates of fire in combat models. Historical analysis has often shown that prescriptive doctrine and rates of fire can vary greatly from what is observed on the battlefield. While doctrine is an important consideration, it should not be the only consideration when a sim-IS environment is intended to simulate how the organizations and personnel are expected to perform.

Recommended Actions(s). Integrated business process conceptual models developed in coordination with the sponsor and/or simulated organization should be evaluated in coordination with the sponsor to determine if they adequately represent how the processes are expected to be executed in the context of the target scenarios. This should include consideration of simulated organizations' adherence to prescribed processes as well as the STS dynamics that may be anticipated in the execution of the processes to be simulated. An example of the former issue may include determining that 10 percent of units are expected to eschew the specified processes and instead use their own internal processes or systems. In such instances where different processes must be simulated for different simulated units or entities, additional conceptual models should be developed to capture the additional processes which are expected to be executed by some simulated units or entities. All relevant STS dynamics should be documented relative to the associated component(s) of the integrated business process conceptual model(s).

b. Activity 2.2: Develop Conceptual Model

> During this activity, the development/integration team produces a conceptual representation of the intended problem space based on their interpretation of user needs and sponsor objectives. The product resulting from this activity is known as a conceptual model... The conceptual model provides an implementation-independent representation that serves as a vehicle for transforming objectives into functional and behavioral descriptions for system and software designers. The model also provides a crucial traceability link between the stated objectives and the eventual design implementation. This model can be used as the structural basis for many design and development activities (including scenario development) and can highlight correctable problems early in the development of the simulation environment when properly validated by the user/sponsor.
>
> The early focus of conceptual model development is to identify relevant entities within the domain of interest, to identify static and dynamic relationships between entities, and to identify the behavioral and transformational (algorithmic) aspects of each entity. Static relationships can be expressed as ordinary associations or as more specific types of

> associations such as generalizations ("is-a" relationships) or aggregations ("part-whole" relationships). Dynamic relationships should include (if appropriate) the specification of temporally ordered sequences of entity interactions with associated trigger conditions. Entity characteristics (attributes) and interaction descriptors (parameters) may also be identified to the extent possible at this early stage of the process. While a conceptual model may be documented using differing notations, it is important that the conceptual model provides insight into the real-world domain and includes an explanatory listing of the assumptions and limitations to properly bound the model.
>
> The conceptual model needs to be carefully evaluated before the next step (design simulation environment) is begun, including a review of key processes and events by the user/sponsor to confirm the adequacy of the domain representation. Revisions to the original objectives and conceptual model may be defined and implemented as a result of this feedback. As the conceptual model evolves, it is transformed from a general representation of the real-world domain to a more specific articulation of the capabilities of the simulation environment as constrained by the member applications of the simulation environment and available resources. The conceptual model will serve as a basis for many later development activities such as member application selection and simulation environment design, implementation, test, evaluation, and validation. (IEEE Std 1730, 2011, p. 15)

Activity 2.2 addresses the development of a sim-IS environment conceptual model. It is important to clearly identify the relationships between the sim-IS environment conceptual model and the other conceptual models influenced by this activity. The sim-IS environment conceptual model is informed by the integrated business model conceptual model. Should RPA be selected as a sim-IS information exchange mechanism, the RPA module conceptual model(s) will be developed in a later activity, informed by the sim-IS environment conceptual model and the conceptual models for the other member applications (simulations and operational ISs) between which they will exchange information.

(1) Issue 2.2.1: Capture Target Integrated Business Processes in the Sim-IS Environment Conceptual Model

Description. Issue 2.1.2 addressed the identification and modeling of relevant integrated business processes which are to be simulated in a sim-IS environment, but it did not address the specification of what component(s) of the process(es) must be represented in the sim-IS environment or scope the degree of fidelity with which they will be simulated. The sim-IS environment conceptual model must identify what components of the target integrated business process(es) are to be represented in the sim-IS environment and the level of fidelity with which they must be simulated.

Recommended action(s). High-level sim-IS environment conceptual models should be developed which identify what components of the integrated business process(es), based on their respective conceptual models, should be represented in the sim-IS environment. This should include specification of any simplifications or abstractions of the integrated business process conceptual models and all STS dynamics to be simulated relative to the appropriate components of the integrated business processes.

(2) Issue: 2.2.2: Define STS Dynamics to be Represented in the Sim-IS Environment

Description. Having identified the integrated business processes to be simulated in the sim-IS environment (Issue 2.2.1) and having identified the STS dynamics for the relevant integrated business processes (Issue 2.1.3), the sim-IS environment designer must next identify which STS dynamics should be simulated in the sim-IS environment and how. This issue should be informed by the purpose for the sim-IS environment and the requirements established in coordination with the sponsor (Issue 1.1.3).

Recommended Actions(s). The sim-IS environment conceptual model should be annotated to identify what STS dynamics are to be simulated for each of the respective integrated business processes components.

(3) Issue 2.2.3: Model Procedures for Information Entry into Respective Operational ISs

Description. While a sim-IS environment conceptual model should be platform-independent relative to the simulation software, it must capture the target integrated business process(es) with a level of detail sufficient to support their simulation in accordance with the sim-IS environment requirements. For some operational ISs, the way information is entered into the IS may impact the way it is presented for the requisite sim-

IS environment participant perspective(s). Low-level conceptual models may be necessary for capturing IS information entry procedures to inform the design of sim-IS information exchange mechanisms simulating the execution of those procedures, including RPA modules.

Recommended action(s). For operational ISs represented in the sim-IS environment, if the mechanism or procedure through which information is entered into the operational IS influences the way it is presented to the sim-IS environment participant, then low-level sim-IS environment conceptual models must also be developed. The low-level conceptual models should align with the high-level sim-IS environment conceptual models while also being precise in their specification of GUI elements with which users interact and any syntax requirements or semantic considerations for the information entered into the operational ISs. While the four levels of UI components presented by Sousa et al. (2010) are likely sufficient for describing user interactions with operational IS GUIs, adding another level of detail in the form of a screen element indicator (discussed in greater detail in Issue 3.3.3) would provide greater specificity for RPA module design and development. These low-level sim-IS environment conceptual models will guide the design of RPA modules (and development of the associated RPA module conceptual models) for the entry of information into the requisite operational ISs.

c. Activity 2.3: Develop Simulation Environment Requirements

> As the conceptual model is developed, it will lead to the definition of a set of detailed requirements for the simulation environment. These requirements, based on the original objectives statement (Step 1), should be directly testable and should provide the implementation level guidance needed to design and develop the simulation environment. The requirements should consider the specific execution management needs of all users, such as execution control and monitoring mechanisms, data logging, etc. Such needs may also impact the scenario developed in Activity 2.1, see 4.2.1. The simulation environment requirements should also explicitly address the issue of fidelity, so that fidelity requirements can be considered during selection of simulation environment member applications. In addition, any programmatic or technical constraints on the simulation environment should be refined and described to the degree of detail necessary to guide implementation activities. (IEEE Std 1730, 2011, p. 16)

(1) Issue 2.3.1: Define requirements for presentation of information for different sim-IS environment participant perspectives

Description. At this point, the sim-IS environment designer should understand how all requisite information must be presented for the different sim-IS environment perspectives and the requisite characteristics of the information to be presented. Requirements for presentation of information for different sim-IS environment participant perspectives should address the hardware and software necessary for presenting the information in a contextually appropriate format, as well as the requirements for the underlying business processes which must be simulated for the presentation of the information and any associated information syntax and semantics requirements.

Recommendation action(s). Requirements should be specified for the form and function of the mediums through which information will be presented for each sim-IS environment participant perspective. This should include any requirements regarding the types of information which must be presented, the processes with which said information must be associated, user actions which must be supported (e.g., drilling down into the details of a maintenance or supply report), specific requirements for how particular pieces of information are represented (e.g., requirements for unique identifiers for entity instances), and relationships between pieces of information (e.g., relationships between equipment maintenance work order numbers and associated repair part requisition numbers). This should also include any requirements for particular procedures which must be followed for the entry of information into respective information presentation mediums like operational ISs.

(2) Issue 2.3.2: Define requirements for simulation of STS dynamics

Description. Requirements should be specified for the STS dynamics which must be simulated for each piece of information relative to the information presentation medium through which it will be presented for sim-IS environment participants.

Recommendation action(s). For each piece of information to be exchanged between a simulation and an IS, any STS dynamics which must be applied to modify the content or timing of the delivery of said information must be specified. This should include identification of requisite sources of STS dynamics, whether they be authoritative sources for real-world dynamics being simulated or a representation of STS dynamics tailored for the purposes of the sim-IS environment.

(3) Issue 2.3.3: Define performance requirements for sim-IS information exchange mechanism(s)

Description. For each piece of information to be exchanged, there may be different requirements for how the information is exchanged and how well the STS dynamics are simulated. Unlike issue 2.3.2, which addresses the identification of STS dynamics which must be simulated through sim-IS information exchange, this issue addresses the specification of requirements for how sim-IS information exchange is conducted. This should include identification of the amount of information that will need to be exchanged and the speed, precision, and frequency with which it must be exchanged, as these requirements will impact the selection of the sim-IS information exchange mechanism(s) in step 3.

Recommendation action(s). In addition to specifying STS requirements which must be simulated, requirements should be provided regarding the performance of the sim-IS information exchange mechanism(s). These requirements should include the amount of information that must be exchanged, the speed, accuracy, and precision with which it must be presented to the requisite IS(s), and the precision with which each of the specified STS dynamics must be simulated.

(4) Issue 2.3.4: Define collection plan for documenting user procedures for interacting with IS GUIs to inform the design of RPA modules

Description. Even with defined procedures for leveraging ISs, users within different departments or units of an organization may employ slightly different procedures for interacting with IS GUIs. Sim-IS environments provide an opportunity for observing and documenting the variety of ways ISs are leveraged in support of common organizational tasks. Such documentation of user processes may prove valuable for informing the design of an organization's standard practices. It may also identify sources of variability in how information is presented to different staff members and decision-makers. Such variability should be identified so that the frequency of its occurrence can be determined for potential simulation in future sim-IS environments.

Recommendation action(s). A collection plan should be developed for documenting the different procedures employed by sim-IS environment participants when interacting with different IS GUIs. These procedures should be modeled and compared with existing RPA conceptual models for the associated task(s) to determine whether the RPA module accurately represents the process or if the observed user-GUI interaction represents a variation in how personnel within an organization leverage their operational IS for the particular task.

(5) Issue 2.3.5: Define collection plan for documenting observed STS dynamics to inform future simulation of STS dynamics

Description. Sim-IS environments may present a rare opportunity for observing and documenting STS dynamics for individuals or unit staffs participating in a sim-IS environment. Observing and documenting STS dynamics like human error and latency for information processed by sim-IS participants provides an opportunity to inform the STS dynamics simulated in future sim-IS environments. These environments present data collection opportunities in operational conditions short of war or other crises.

Recommendation action(s). A collection plan should be developed for the documentation of STS dynamics observed among sim-IS environment participants. These observed STS dynamics should be leveraged to inform the continuous refinement of the database of STS dynamics to be represented in sim-IS environments and the design of RPA modules which may be employed to support simulation of said STS dynamics. Conceptual models for the work processes should be annotated to associate the STS dynamics (e.g., data entry errors or latency) with the step in the associate integrated business process. Such documentation of STS dynamics should also include characteristics of the personnel or staff from which they originated. This should include information such as how long the staff has been training together and their cumulative experience, which may assist in determining whether the STS dynamics are representative of the broader community.

(6) Issue 2.3.6: Define requirements for management of RPA modules

Description. RPA software can be employed from local computer desktops or remote RPA servers. They can be run in attended mode or unattended. Prior to designing the sim-IS environment and selecting the appropriate information exchange mechanism(s), the sim-IS environment designer must identify any organizational limitations or requirements for the employment of RPA software.

Recommendation action(s). RPA software employment requirements should be identified in coordination with the definition of execution management requirements, software/hardware requirements, and security requirements. RPA considerations should also be included in the specification of sim-IS environment test criteria to ensure the wide variety of likely interactions between simulations and ISs through RPA are tested in advance of executing RPA-based sim-IS environments.

3. Step 3: Design Simulation Environment

> The purpose of this step of the DSEEP is to produce the design of the simulation environment that will be implemented in Step 4. This involves identifying applications that will assume some defined role in the simulation environment (member applications) that are suitable for reuse, creating new member applications if required, allocating the required functionality to the member application representatives, and developing detailed planning documents [sic]. (IEEE Std 1730, 2011, p. 17)

Figure 59 illustrates the activities associated with Step 3 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

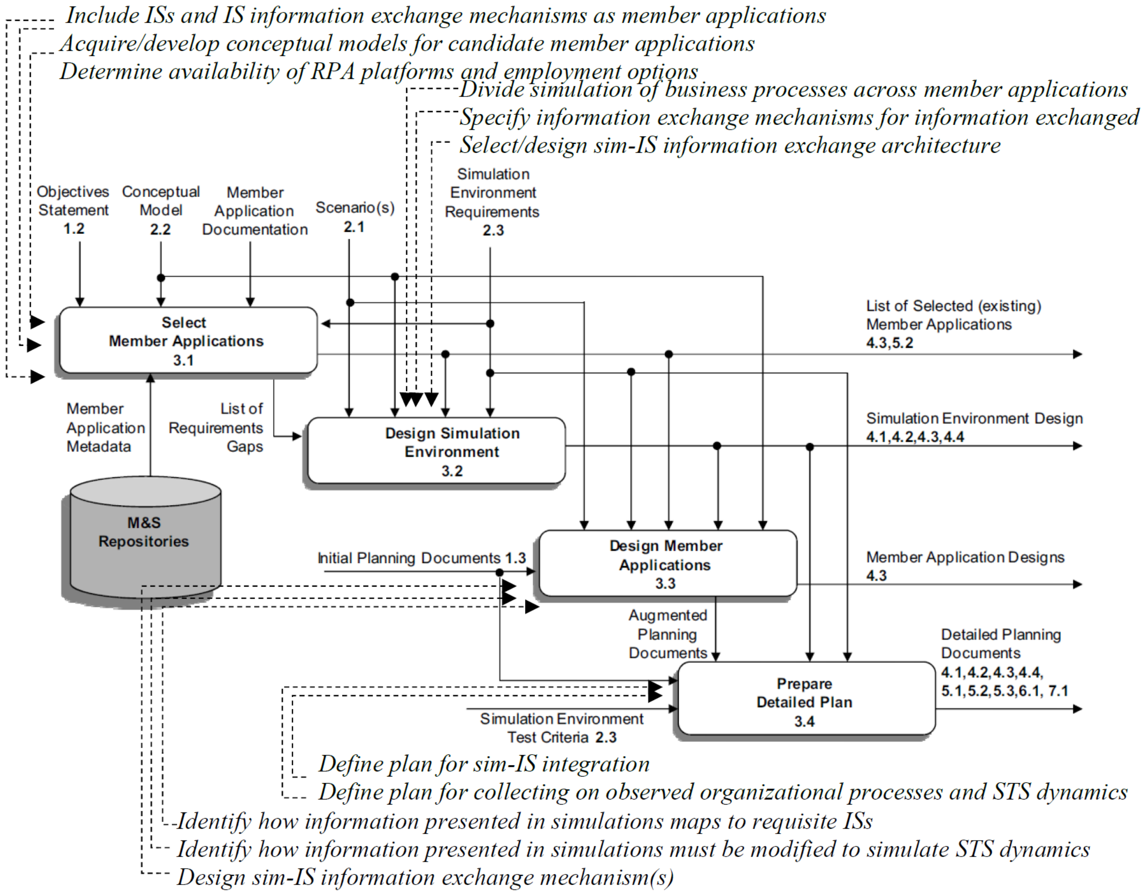


Figure 59. DSEEP Step 3: Design Simulation Environment. Source: IEEE Std 1730 (2011) © 2011 IEEE.

a. *Activity 3.1: Select Member Applications*

The purpose of this activity is to determine the suitability of individual simulation systems to become member applications of the simulation environment. This is normally driven by the perceived ability of potential member applications to represent entities and events according to the conceptual model. In some cases, these potential member applications may be simulation environments themselves, such as an aircraft simulation built from separate simulations of its subsystems. Metadata describing reusable and available member applications may be leveraged to discover candidate assets within existing M&S repositories. Managerial constraints (e.g., availability, security, facilities) and technical constraints (e.g., VV&A status, portability) may both influence the final selection of member applications.

In some simulation environments, the identity of at least some member applications will be known very early in the process. For instance, the sponsor may explicitly require the use of certain member applications in the simulation environment, or an existing simulation environment (with well-established member applications) may be reused and extended as necessary to address a new set of requirements. Although early member application selection may have certain advantages, it also introduces some immediate constraints on what the simulation environment will and will not be able to do. Since required capabilities are not always well understood at the initiation of the development activities, it is generally advisable to defer final decisions on simulation environment membership until this point in the overall process.

In some situations, it may be possible to satisfy the full set of requirements for the simulation environment with a single simulation. The use of a single simulation (with modifications as necessary) would eliminate many of the development and integration activities required for multi-member distributed simulation environments (as described in Steps 4 and 5). The developer/integrator should compare the time and effort required to perform the necessary modifications against the time and effort required to assimilate an established set of member applications into an integrated simulation environment. Other factors, such as reusability of the resulting software, should also be taken into account in deciding the proper design strategy.

Existing repositories should be searched for candidate member applications, keyed to critical entities and actions of interest. To support final member application selection decisions, additional information resources (such as design and compliance documents) are generally necessary to fully understand internal simulation representations of required behaviors/activities and other practical use constraints. Finally, it may not be possible to make a firm decision between two competing member applications with the available data. If this situation arises, then both member applications may be taken to the design activity that follows to perform more detailed analysis and testing during the design activity. (IEEE Std 1730, 2011, p. 18)

(1) Issue 3.1.1: Include Selection of ISs and Sim-IS Information Exchange Mechanisms as Sim-IS Environment Member Applications

Description. This overlay is based on the premise that applying DSEEP to sim-IS environments requires expanding the scope of the simulation environment to include both operational ISs to be populated and the means through which they are populated (sim-IS information mechanisms). This necessitates considering both operational ISs and sim-IS information exchange mechanisms as sim-IS environment member applications.

Recommended action(s). In all DSEEP activities where member applications are referenced (in DSEEP steps 3 through 6), member applications should be understood to refer not only to simulations, but also the operational ISs being stimulated or emulated, and the sim-IS information exchange mechanisms being employed to simulate the STS processes through which they would be populated in the real world. For Activity 3.1, sim-IS environment designers should therefore define member application selection criteria for operational ISs (or emulators) and sim-IS information exchange mechanisms (activity task 3.1.2.1), search repositories for IS and sim-IS information exchange mechanisms (activity task 3.1.2.3), and analyze the ability of ISs and sim-IS information exchange mechanisms to "represent required entities, events, and phenomena" (IEEE Std 1730, 2011, p. 19), including associated processes and STS dynamics (activity task 3.1.2.5). While the allocation of functions to member applications is not conducted until the next activity (activity 3.2), the sim-IS designers should identify initial reasons for the selection of sim-IS environment member applications, including how STS processes to be simulated are expected to be allocated across simulations, operational ISs, and associated sim-IS information exchange mechanisms.

(2) Issue 3.1.2: Acquire or Develop Conceptual Models for Candidate Member Applications

Description. Selection of sim-IS environment member applications should be supported by comparing conceptual models for the candidate member applications (simulations, ISs, and sim-IS information exchange mechanisms) to the sim-IS environment conceptual model. Selection of appropriate simulations in this activity is generally considered as applying to the selection of simulations in a conventional sense, with interoperability mechanisms addressed later in the development step of DSEEP. For sim-IS environments where the conceptual model captures the nature of information progression from the ground truth to ISs through STSs, the selection of sim-IS information exchange mechanisms should be included in this design activity. This can include modeling sim-IS information exchange through conventional means, routing sim-IS information exchange through communications effects servers for simulation of propagation effects and network effects, or the use of RPA-based approaches for simulation of STS and cyber effects not supported by communications effects servers.

Just as conceptual models are necessary for defining what is to be simulated in the sim-IS environment, conceptual models of candidate member applications are necessary for determining whether candidate member applications will adequately simulate the requisite entities, objects, and phenomena. Conceptual models for simulations, ISs, and sim-IS information exchange mechanisms are necessary for informing sim-IS environment designers regarding how exactly the objects, events, and phenomena are represented in each candidate member application and to compare those representations to the desired representations specified in the sim-IS environment conceptual model.

Recommended action(s). Sim-IS environment designers should acquire or develop conceptual models for each of the candidate member applications, including the entities/objects and functions which they are expected to be responsible for in the sim-IS environment. This should include conceptual models for simulations, sim-IS information exchange mechanisms, and IS emulators being considered. Conceptual models for operational ISs are necessary as well. If a variant of an operational IS is being considered as a candidate member application, for example, a conceptual model of that IS should be compared with the conceptual model of the real-world operational IS to ensure it adequately represents the presentation of information and the functionality of the operational IS in accordance with the sim-IS environment requirements.

(3) Issue 3.1.3: Determine Availability of RPA Platforms and Options for Employment

Description. The number of RPA platforms that are available may impact the feasibility of using RPA-based sim-IS information exchange for some or all of the sim-IS information exchange requirements. As previously discussed, the number of RPA bot instances necessary for supporting sim-IS information exchange must be evaluated based on the speed of the RPA workflows, the frequency and timeliness required for the sim-IS information exchange, and the number of instances of information which must be exchanged between simulations and ISs. While the sim-IS environment designer can leverage the processes previously presented to determine the requisite number of RPA bot instances, they must know the number of RPA bot instances which are available to them for the sim-IS environment when determining the suitability of different sim-IS information exchange mechanisms.

Recommended action(s). Identify the number of RPA platforms which are available for use in the sim-IS environment, including availability of RPA modules for each of the simulations and operational ISs being considered as member applications. This should include identification of whether the RPA instances will be hosted on local computers or on a centralized server, and identification of whether the organization has authorization to run RPA instances in attended and/or unattended modes under the intended level of classification for the scenario and with each of the other member applications.

b. Activity 3.2: Design Simulation Environment

> Once all member applications have been identified, the next major activity is to prepare the simulation environment design and allocate the responsibility to represent the entities and actions in the conceptual model to the member applications. This activity will allow for an assessment of whether the set of selected member applications provides the full set of required functionality. A by-product of the allocation of functionality to the member applications will be additional design information that can embellish the conceptual model.
>
> A fundamental design choice for any distributed simulation environment is the underlying simulation architecture (e.g., HLA, DIS, TENA). In some cases, the requirements of the application will align with only one such architecture. In other cases, any of several different architectures could potentially satisfy the simulation environment requirements. In this latter case, several factors must be taken into account in making the final selection, such as the degree of training and experience of the development/ integration team on each of the candidate architectures, adequacy of supporting data models, robustness and performance of the architecture middleware, and the availability and affordability of other required resources (e.g., tools, documentation)…
>
> In some large simulation environments, it is sometimes necessary to mix several simulation architectures. This poses special challenges to the simulation environment design, as sophisticated mechanisms are sometimes needed to reconcile disparities in the architecture interfaces. For instance, gateways or bridges to adjudicate between different on-the-wire protocols

> are generally a required element in the overall design, as well as mechanisms to address differences in SDEMs. Such mechanisms are normally formalized as part of the member application agreements, which are discussed in Step 4.
>
> As agreements on assigned responsibilities are negotiated, various design trade-off investigations may be conducted as appropriate to support the development of the simulation environment design. Many of these investigations can be considered to be early execution planning and may include technical issues such as time management, execution management, infrastructure design, runtime performance, and potential implementation approaches.
>
> The major inputs to this activity include the simulation environment requirements, the scenario, and the conceptual model… In this activity, the conceptual model is used as a conduit to make sure that user domain-specific requirements are appropriately translated into the simulation environment design. High-level design strategies, including modeling approaches and/or tool selection, may be revisited and renegotiated at this time based on inputs from the member application representatives. When the simulation environment represents a modification or extension to a previous simulation environment, new member application representatives must be made cognizant of all relevant negotiated agreements and given the opportunity to revisit pertinent technical issues. For secure applications, efforts associated with maintaining a secure posture during the simulation execution can begin, including the designation of security responsibility. The initial security risk assessment and concept of operations may be refined at this time to clarify the security level and mode of operation. (IEEE Std 1730, 2011, p. 20)

(1) Issue 3.2.1: Identify How Simulation of Integrated Business Processes Is Divided Across Member Applications

Description. During this activity, DSEEP recommends that sim-IS designers "identify [selected members] that best provide required functionality and fidelity" (IEEE Std 1730, 2011, p. 21) and "allocate the responsibility to represent the entities and actions in the conceptual model to the member applications" (IEEE Std 1730, 2011, p. 19). For the simulation of integrated business processes through sim-IS environments, the allocation of responsibilities across member applications should also include identification of how sim-IS information exchange supports simulation of the target integrated business process(es) as specified in the sim-IS environment conceptual model.

Recommended action(s). For sim-IS environments, the allocation of functionality to member applications should be extended to include identifying what portions of the sim-IS conceptual model are simulated through sim-IS information exchange mechanisms and how. Where puckster-based sim-IS information exchange is employed, the procedures through which sim-IS information exchange occurs should be aligned with the corresponding component(s) of the sim-IS environment conceptual model. For RPA-based sim-IS information exchange, the identification of which portions of the sim-IS environment conceptual model are simulated by the simulation, which ones are covered by the RPA middleware, which ones are covered by the operational IS or emulator, and how they all relate to each other, will inform the design of the RPA module(s) and support the module(s) management in case of future changes to the sim-IS environment, simulation, or operational IS.

(2) Issue 3.2.2: Identify Sim-IS Information Exchange Mechanism for Each Type of Information to Be Exchanged

Description. Simulating the processes through which different types of information propagate across STSs may necessitate the use of different sim-IS information exchange mechanisms. Requirements for the simulation of integrated business processes should be defined earlier in DSEEP (Activity 2.3), informing the selection of member applications (Activity 3.1). Identification of the appropriate sim-IS information exchange mechanism for each information item in Activity 3.2 should be informed by weighing the requirements against the strengths and weaknesses of the different sim-IS information exchange mechanisms. Simulating GPS signals, for example, may be better suited for protocol-based or ad hoc point-to-point sim-IS information exchange rather than puckster or RPA-based sim-IS information exchange, due to the lower speeds of information exchange afforded by the latter two mechanisms. If modification of information is required for simulation of STS dynamics, sim-IS designers could couple the protocol-based approach with a communications effects server or design an ad hoc point-to-point connection to simulate the requisite dynamics.

Recommended action(s). Sim-IS environment designers should identify the sim-IS information exchange mechanism to be employed for effecting information exchange for each type of information to be exchanged. For each type of information this should include identification of the information source (simulation), recipient (IS), and any translation and/or degradation of the information that must be conducted by the sim-IS information exchange mechanism. This should include identifying where information is extracted from (in simulations), where it is entered into (in ISs), and how these relate to the integrated business process(es) as represented in the sim-IS environment conceptual model. It should also include identification of what STS dynamics are simulated by each member application, as represented in the sim-IS environment conceptual model.

(3) Issue 3.2.3: Design Sim-IS Information Exchange Infrastructure

Description. DSEEP Activity 3.2 includes a recommendation to "develop design of the simulation environment infrastructure, and select protocol standards and implementations" (IEEE Std 1730, 2011, p. 21). For sim-IS environments this should include design of sim-IS information exchange mechanisms and the associated architectures, where applicable. Following identification of which sim-IS information exchange mechanism is best suited for each sim-IS information exchange requirement (Issue 3.2.2), an initial high-level architecture design should be developed for each sim-IS information exchange mechanism to inform the design of member applications (Activity 3.3). This can include the design of the architecture for RPA-based sim-IS information exchange at a high-level (Issue 3.3.2). As member applications, including sim-IS information exchange mechanisms, are designed in the next activity (Activity 3.3), sim-IS environment designers may need to revisit this issue, adjusting the design of the sim-IS information exchange mechanism architecture. This activity can also include the design of processes and procedures through which sim-IS information exchange is achieved with other sim-IS information exchange mechanisms, including procedures for manual actions by pucksters.

Recommended action(s). The architecture for each sim-IS information exchange mechanism should be designed at a high-level. This should include identification of how each sim-IS information exchange mechanism will support the translation and modification of information exchanged between simulations and ISs, as required. The relationships between sim-IS information exchange mechanisms and supporting databases should also be identified, including requirements for databases to support translation in dynamic sim-IS information exchange and requirements for synchronized StartEx data.

c. *Activity 3.3: Design Member Applications*

> In some circumstances, an existing set of member applications cannot fully address all defined requirements for the simulation environment. In these cases, it may be necessary to perform an appropriate set of design activities at the individual member application level. This may involve enhancements to one or more of the selected member applications, or could even involve designing an entirely new member application. The purpose of this activity is to transform the top-level design for the simulation environment into a set of detailed designs for the member applications. The scope of the design task will depend on the amount of previous design work that can be reused. New member applications will generally require a substantial amount of design effort whereas modifications to existing member applications will require less effort. When existing member applications are being modified, it is necessary to document any changes to facilitate good configuration control.(IEEE Std 1730, 2011, p. 22)

(1) Issue 3.3.1: Identify How Information Presented in Selected Simulations Maps to Requisite ISs

Description. During Activity 1.2, requirements should have been specified for what information needs to be represented in each IS and how (Issue 1.2.1). With the member applications having been selected in Activity 3.1 and functionality assigned to member applications in Activity 3.2, the sim-IS designer should be prepared in Activity 3.3 to identify how information needs to be translated from how it is presented at its source simulation to the operational IS (or other medium) through which it is presented for the appropriate sim-IS environment participant perspective(s).

Recommended action(s). Identify how information generated by simulations maps to the required representations in operational ISs to inform the design of requisite sim-IS environment databases and sim-IS information exchange mechanisms. Using the requirements specified in Activity 1.2, the sim-IS environment designer should identify how the information generated by simulations to be presented in different operational ISs (or other information presentation mediums) needs to be translated to meet IS syntax and semantics requirements. This can include identifying how entity characteristic categories specified by a simulation relate to corresponding IS categories (e.g., equipment readiness or personnel casualties). This activity should also identify how to resolve differences in levels of aggregation for information exchanged between simulations and operational ISs. For entities like equipment and personnel, this can include identification of how unique identifiers are tracked and how changes in aggregate equipment and personnel statuses are represented as individual entity status changes. For simulation of characteristics of objects like supplies, this can include addressing issues like the translation of aggregate fuel to bulk fuel statuses, as discussed in Chapter I.

Defining how simulation-generated information maps to how information is presented in the requisite operational IS may require revisiting Activity 1.2 to ensure the manner of translation from simulation information to presentation in operational ISs meets the sponsor's requirements.

(2) Issue 3.3.2: Identify How Information Presented in Selected Simulations Must Be Modified to Simulate Target STS Dynamics

Description. The simulation of STS dynamics may be achieved in a sim-IS environment through functionality provided by one or multiple simulation systems and/or sim-IS information exchange mechanisms. After selecting the sim-IS environment member applications and allocating sim-IS environment functionality to each (including identifying the sim-IS information exchange mechanisms for each piece of information to be exchanged), the sim-IS environment designer is positioned to identify the role of each member application in the modification of simulation-generated information for the simulation of target STS dynamics. This process of identifying each member application's role in simulating STS dynamics is a necessary first step for identifying requisite changes for simulations as well as informing the design of the sim-IS information exchange mechanism.

Recommended action(s). Identify how information generated in sim-IS environment simulations will be modified by simulations and/or sim-IS information exchange mechanisms to achieve simulation of the requisite STS dynamics.

(3) Issue 3.3.3: Design Sim-IS Information Exchange Mechanisms

Description. After identifying the role of each member application in supporting the translation and modification of simulation-generated information (Issues 3.3.1 and 3.3.2, respectively), the sim-IS environment designer is positioned to design sim-IS information exchange mechanism(s) to simulate target integrated business processes and simulate requisite STS dynamics. Where an RPA-based sim-IS information exchange approach is employed, multiple levels of conceptual models are necessary: the sim-IS environment conceptual model and conceptual models for the individual RPA modules supporting the sim-IS information exchange. While the design of all RPA modules must be considered together for simulating the phenomena represented in the sim-IS environment model, certain RPA modules are particularly responsible for addressing issues 3.3.1 and 3.3.2, as depicted in Figure 60.

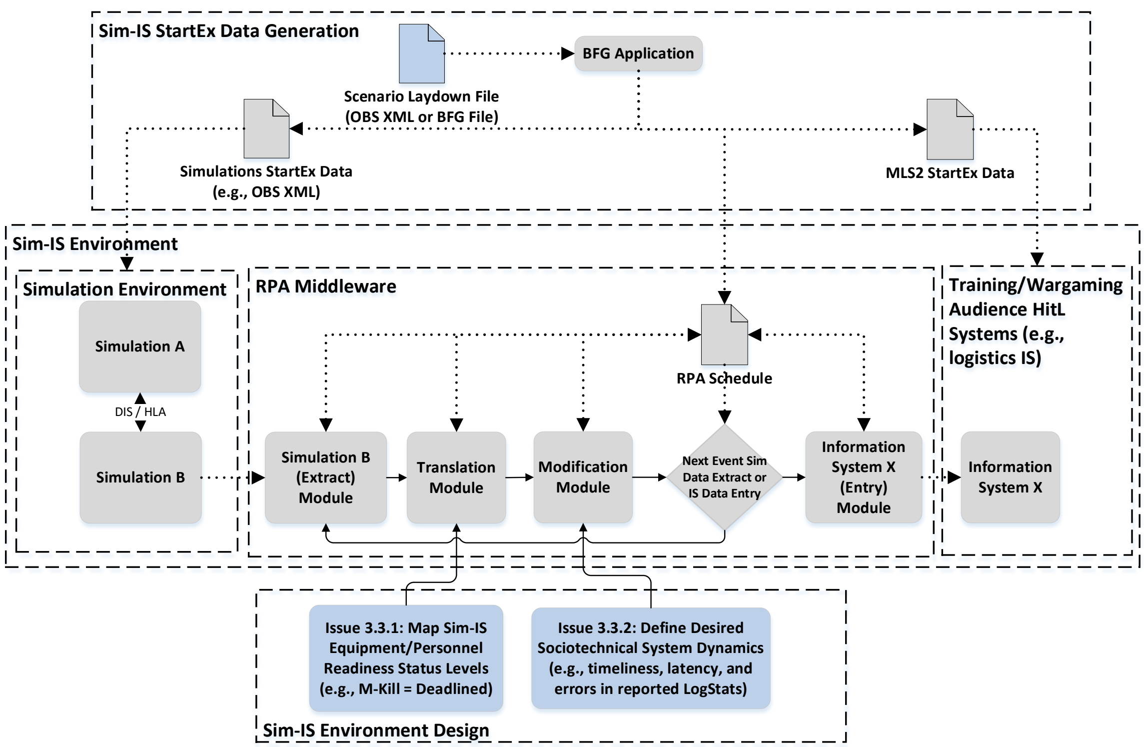


Figure 60. Design of RPA Modules for Sim-IS Information Exchange to Address Issues 3.3.1 and 3.3.2.

The sim-IS environment conceptual model identifies how integrated business processes and STS dynamics are to be simulated. The RPA modules' conceptual models should capture how RPA workflows will simulate the target integrated business processes and STS dynamics in coordination with the other member applications and in accordance with the sim-IS environment conceptual model. This should include detailed modeling of how RPA module(s) will simulate human interactions with operational ISs' user interfaces. This activity should also address the design of individual RPA modules to address Issues 3.3.1 and 3.3.2 for simulation of integrated business processes and STS dynamics. While RPA module design should address how RPA modules will receive information from simulations, detailed modeling of interactions with simulation interfaces falls outside the scope of the design step and should instead be addressed in the DSEEP step 4 for sim-IS environment development.

Sousa et al. (2010) present four levels of UI components to support UI modeling: screen group, screen, screen fragment, and screen element. While these levels may be sufficient for modeling human interactions with UIs, an additional level of abstraction is necessary for modeling RPA workflow interactions with UIs: screen element indicator. RPA workflows use a variety of indicators or anchors to support identification of appropriate screen elements for interaction. The screen element indicator would identify the indicator(s) which may be used by an RPA workflow to identify screen elements, with a specified degree of confidence, and possibly through the use of specified variables.

Recommended action(s). After addressing Issues 3.3.1 and 3.3.2, design the sim-IS information exchange mechanism(s) to support the translation and modification of information as it is exchanged from simulation(s) source to requisite IS(s) or other medium to inform each sim-IS environment participant perspective. This should include designing the sim-IS information exchange mechanism(s) to support simulation of target integrated business processes and associated STS dynamics. Where the RPA-based approach has been selected for sim-IS information exchange, this should include initial identification of the sim-IS data transformation component files which will need to be developed as part of the RPA-based sim-IS information exchange architecture, and the simulation and IS parametric data with which these files will need to be aligned and in what ways. This will inform not only the design of the RPA modules, but also the sim-IS data exchange requirements (Issue 4.1.1) and the StartEx data generation and synchronization process.

Two levels of conceptual models should also be developed concurrently with the design of the RPA modules: sim-IS environment conceptual model and RPA modules' conceptual models. The former should include identification of the specific components of IS interfaces with which the associated RPA modules interact, specified down to the specific screen element and the screen element indicators which are reliable enough for use with RPA modules. The conceptual models should be maintained together to facilitate the maintenance and modernization of the RPA modules as the simulations, ISs, and simulated processes evolve over time. Maintenance of these sets of conceptual models may facilitate thorough but efficient evaluation of the suitability of RPA modules for reuse in support of future sim-IS environments.

(4) Issue 3.3.4: Designing for Schedule or Event-based Sim-IS Information Exchange

Description. The architecture presented in this research for RPA-based sim-IS information exchange does not necessarily support all sim-IS information exchange requirements. One type of sim-IS information exchange which may require a different RPA-based sim-IS information exchange architecture is event-based sim-IS information exchange. Information exchanged across sociotechnical systems can include both scheduled exchanges (e.g., daily logstat updates) and event-based exchanges. Event-based information exchanges include responses to requests for information and information transmitted based on CCIRs. While responding to requests for information includes natural language processing, which is outside the scope of this research, certain CCIR reports may be simulated through an RPA-based sim-IS environment. While this research focused on the simulation of STS dynamics in recurring STS processes such as status updates, event-based sim-IS information exchange may necessitate additional design considerations.

Recommended action(s). During the design of the RPA-based sim-IS information exchange mechanism, the sim-IS environment designer should determine if one RPA-based sim-IS environment architecture is sufficient for supporting all information exchange requirements. If different RPA-based sim-IS information exchange architectures are required, they should determine whether RPA modules (e.g., simulation information extraction or IS information entry modules) can be reused across different architectures.

d. Activity 3.4: Prepare Detailed Plan

> Another major activity in Step 3, design simulation environment, is to develop a coordinated plan to guide the development, test, and execution of the simulation environment. This requires close collaboration among all member application representatives to obtain a common understanding of the various goals and requirements and also to identify (and agree to) appropriate methodologies and procedures based on recognized systems engineering principles. The initial planning documents prepared during development of the original objectives provide the basis for this activity... The plan should include the specific tasks and milestones for each member application, along with proposed dates for completion of each task.
>
> The plan may also identify the software tools that will be used to support the remaining life cycle of the simulation environment (e.g., CASE, configuration management, VV&A, testing). For new simulation environments, a plan to design and develop a network configuration may be required. These agreements, along with a detailed work plan, must be documented for later reference and possible reuse in future applications.(IEEE Std 1730, 2011, p. 22)

(1) Issue 3.4.1: Define Plan for Sim-IS Integration

Description. The plan developed within Activity 3.4 for the integration of the simulation environment should include planning for the integration of simulations and ISs through the selected sim-IS information exchange mechanisms. If the RPA-based approach is one of the selected sim-IS information exchange mechanisms, this integration plan should include the process through which the requisite number of RPA bot instances will be determined.

Recommended action(s). Define a plan for achieving the integration of requisite pairs of simulations and ISs through sim-IS information exchange mechanisms as designed in the previous activity. This should include a plan for the development of the sim-IS StartEx data generation and synchronization process, including a process for identifying all simulation and IS parametric data which must be configured to support RPA-based sim-

IS information exchange and how to align the requisite components of the parametric data accordingly.

(2) Issue 3.4.2: Define Plan for Collecting Data on Observed Organizational Processes, STS Dynamics, and User-GUI Interactions

Description. The design of sim-IS environments to simulate operational integrated business processes depends on an understanding of those processes and how they are executed in context. As sim-IS environments are leveraged to support training and evaluation of organizational processes and decision-making in context, they also provide an opportunity to observe some of those same processes in context. By collecting on the observed processes and STS dynamics, sim-IS environment designers can conduct iterative development of the sim-IS environment and coevolve with the integrated business processes they are supporting.

Recommended action(s). Define a collection plan to document what is observed in trainee actions, in order to inform the continuous refinement of how the sim-IS environment represents those actions and associated dynamics. This should include a plan for documenting and modeling organizational processes, STS dynamics, and user interactions with IS GUIs. It can consist of documentation conducted by trained research or training personnel as well as the instrumentation of training environments and systems for automated collection.

4. Step 4: Develop Simulation Environment

> The purpose of this step of the DSEEP is to define the information that will be exchanged at runtime during the execution of the simulation environment, modify member applications if necessary, and prepare the simulation environment for integration and test (database development, security procedure implementation, etc.).(IEEE Std 1730, 2011, p. 24)

Figure 61 illustrates the activities associated with Step 4 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

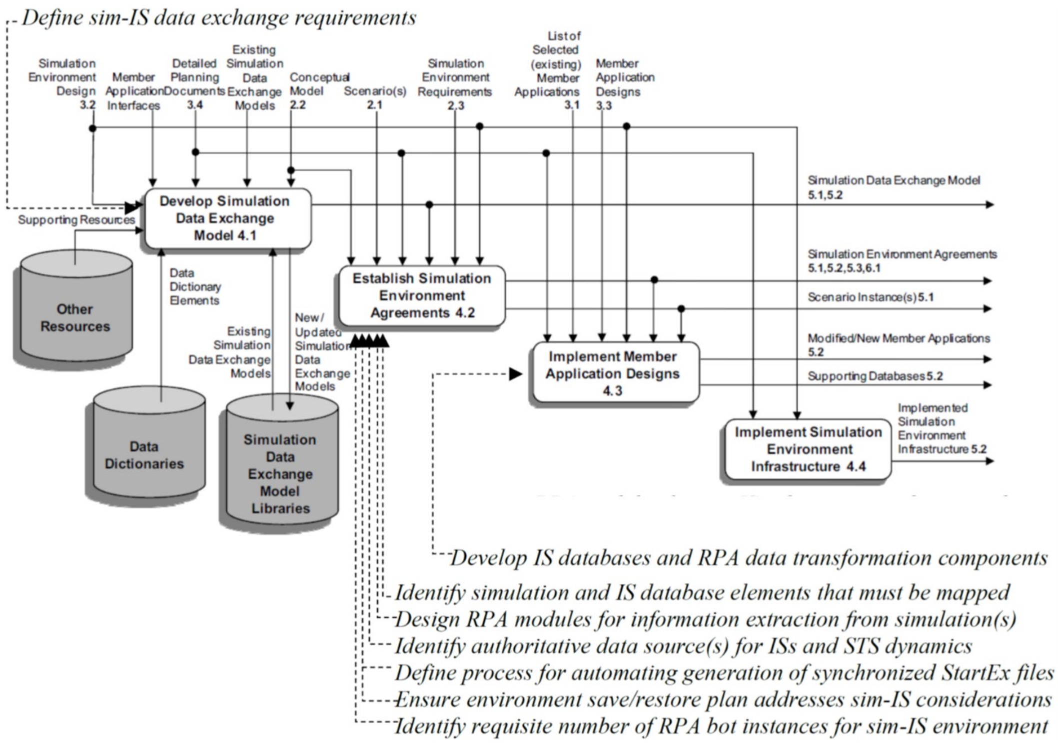


Figure 61. DSEEP Step 4: Develop Simulation Environment. Source: *IEEE Std 1730* (2011) © 2011 IEEE.

a. Activity 4.1: Develop Simulation Data Exchange Model

In order for the simulation environment to operate properly, there must be some means for member applications to interact. At a minimum, this implies the need for runtime data exchange, although it could also involve remote method invocations or other such means of direct interaction among cooperating object representations. Clearly, there must be agreements among the member applications as to how these interactions will take place, defined in terms of software artifacts like class relationships and data structures. Collectively, the set of agreements that govern how this interaction takes place is referred to as the simulation data exchange model (SDEM).

Depending on the nature of the application, the SDEM may take several forms. Some simulation applications are strictly object-oriented, where both static and dynamic views of the simulation system are defined in terms of class structures, class attributes, and class operations (i.e., methods) within the SDEM. Other simulation applications maintain this same object-based paradigm, but use object representations as a way to share state information among different member applications about entities and events that are being modeled internal to the member applications themselves. In this case, the SDEM is quite similar to a data model, including a defined set of format, syntax, and encoding rules. Still other simulation applications may not use an object-based structure at all in their SDEM. Rather, the focus is on the runtime data structures themselves and the conditions that cause the information to be exchanged. In general, different applications will have different requirements for the depth and nature of interaction among member applications, and while these varying requirements will drive the type and content of the SDEM, it is necessary for the SDEM to exist in order to formalize the "contract" among member applications necessary to achieve coordinated and coherent interoperation within the simulation environment.

There are many possible approaches to SDEM development. Certainly, reuse of an existing SDEM is the most expeditious approach, if one can be found within a repository that meets all member application interaction requirements. If an exact match for the needs of the current application cannot be found, then identifying an SDEM that meets some of these needs and modifying/tailoring the SDEM to meet the full set of requirements would generally be preferable to starting from a "clean sheet." Some communities maintain reference SDEMs for their users to facilitate this type of reuse (e.g., the real-time platform reference federation object model (RPR FOM); see SISO-STD-0001.1 [B2]). Still other approaches involve assembling an SDEM from small, reusable SDEM components (e.g., base object models (BOM), see SISO STD-003 [B3]) and merging SDEM elements from the interfaces of member applications [e.g., HLA simulation object models (SOM)]. In general, the simulation environment development/integration team should employ whatever approach makes most sense from a cost, efficiency, and quality perspective. (IEEE Std 1730, 2011, p. 25)

(1) Issue 4.1.1: Define Sim-IS Data Exchange Requirements

Description. In the previous DSEEP steps, the sim-IS designer should have coordinated with the sponsor to identify the information which must be presented for different sim-IS environment participant perspectives, the medium(s) through which it must be presented, and the semantics, syntax, and information entry procedure requirements for populating those mediums, including operational ISs. During Activity 4.1, the sim-IS designer should identify how information generated by the selected sim-IS environment simulations must be adjusted to facilitate its entry into the information mediums as previously defined. This includes identification of how the semantics and syntax of simulation-generated information relates to the semantics and syntax requirements for entry of information into operational ISs. This should also include the identification of how simulation and IS information (and associated databases) must be aligned based on the design of the sim-IS information exchange mechanism(s) defined in Activity 3.3 (Issue 3.3.3).

Recommended action(s). Identify how information received from requisite simulation(s) must be adjusted to address the semantics and syntax requirements (identified in previous DSEEP steps) for its entry into the requisite operational ISs and other mediums for presentation in support of different sim-IS environment participant perspectives. These sim-IS data exchange requirements should be informed by the design of RPA sim-IS information exchange modules, the IS data entry requirements, and analysis of how information is presented in requisite simulations for extraction by different sim-IS information exchange mechanisms. Where RPA-based sim-IS information exchange is employed, this analysis should inform the development of RPA translation modules and the development of simulation and IS databases to ensure they are adequately aligned to support sim-IS information exchange. This may require RPA concurrent design and development of RPA transformation component files to support the mapping of simulation and IS information, such as object class identifiers and entity instance identifiers.

b. Activity 4.2: Establish Simulation Environment Agreements

> Although the SDEM represents an agreement among member application representatives as to how runtime interaction will take place, there are other operating agreements that must be reached that are not documented in the SDEM. Such agreements are necessary to establish a fully consistent, interoperable, simulation environment. While the actual process of establishing agreements among all participants in the development effort begins early in the DSEEP and is embodied in each of its activities, this may not result in a complete set of formally documented agreements. It is at this point in the overall process that the development/integration team needs to explicitly consider what additional agreements are required and how they should be documented. (IEEE Std 1730, 2011, p. 26)

(1) Issue 4.2.1: Identify Simulation and IS Databases and Database Elements That Must Be Consistent or Mapped

Description. To support sim-IS information exchange, information to be presented in simulation(s) must be associated with the corresponding information to be represented in IS(s) being populated. While the previous DSEEP steps identified this information generally and in terms of how it manifests in simulation and IS interfaces, in Activity 4.2 sim-IS developers must identify what simulation and IS databases are associated with the manner through which that information manifests in system interfaces. They must also identify how these databases need to be aligned or mapped to support sim-IS information exchange. Depending on the sim-IS data exchange requirements defined in Activity 4.1 (Issue 4.1.1), the databases may need to include mirrored information or mapped information. This will depend on the design of the sim-IS information exchange mechanism(s). If the sim-IS information exchange mechanism is designed to support information exchange for object classes or entity instances with common identifiers in both simulation and IS GUIs, then certain information will have to be mirrored across different databases. If the sim-IS information exchange mechanism is designed to support information exchange where the object class of entity instances have different identifiers in the GUI, then tables may be required which support the mapping of values across different databases.

Recommended action(s). Identify simulation and IS databases and database elements that must be consistent or mapped to support the associated sim-IS information exchange mechanisms. This should include what elements of the databases must be consistent and any semantics or syntax requirements for the databases. If the database elements must be mapped, this activity should also identify what data elements or tables are to support the mapping of the requisite simulation and IS databases and how they are designed. This activity should also define the process through which requisite databases and database elements of simulations and ISs (and associated sim-IS information exchange mechanism(s) data elements) are updated in case of future changes.

(2) Issue 4.2.2: Design RPA Modules for Information Extraction from Simulation(s)

Description. While the design of RPA modules for the extraction of requisite information from simulations was addressed at a high level in Activity 3.3 (see Issue 3.3.3), detailed modeling of the RPA module(s) for interaction with the requisite simulation interface(s) should be addressed in Activity 4.2. This should complete the design of the

RPA modules for simulation information exchange in coordination with the design of the simulation, IS, and RPA databases and sim-IS data transformation components.

Recommended action(s). This activity should include identification of the specific processes and procedures through which requisite information is extracted from simulation interfaces as part of the broader RPA-based sim-IS information exchange process. This should be informed by the RPA module design developed in Activity 3.3 (Issue 3.3.3), providing an additional level of detail regarding RPA module interactions with requisite simulation interfaces based on sim-IS information mapping addressed in Issue 4.2.1.

(3) Issue 4.2.3: Identify Authoritative Data Sources for ISs and STS Dynamics

Description. Authoritative data sources identified for sim-IS environments should include the source of information represented in operational ISs as well as the data sources informing the simulation of STS dynamics. As the simulated STSs evolve, the STS dynamics should be expected to evolve as well.

Recommended action(s). Identify authoritative data sources for operational ISs included in the sim-IS environment and data sources informing the simulation of STS dynamics. Identify any possible conflict between operational ISs' data sources and the sim-IS environment information to be presented through the operational IS, and methods for mitigating risk.

(4) Issue 4.2.4: Define Process for Generation of Synchronized Simulation and IS Scenario Initialization Files

Description. Sim-IS developers should be prepared to go through the process of generating and synchronizing databases and scenario initialization documents for sim-IS environment member applications (simulations, ISs, and sim-IS information exchange mechanisms) multiple times. They should also be prepared to conduct the process under time constraints, as simulation environments are often modified in the final moments before execution. The process should therefore be clearly defined in Activity 4.2 to support the efficient and effective modification of all requisite databases, data transformation components, and other StartEx files as required. Tools for automating the generation of synchronized StartEx files should be identified. These can include the JTDS for generating a common OBS for populating simulations and the BFG application for generating MLS2 databases aligned with the OBS. It should also include detailed instructions for the modification of requisite parametric data elements to support sim-IS information exchange, as seen in Appendix D.

Recommended action(s). Following identification of all requirements for the design and alignment of simulation, IS, and sim-IS information exchange mechanism databases (Issue 4.2.1), the process for generating and synchronizing the requisite databases and scenario initialization files should be defined. This should include the identification of any required tools or templates and their role in the process.

(5) Issue 4.2.5: Ensure the Plan for Saving or Restoring the Environment Includes Sim-IS Information Exchange Considerations

Description. Activity 4.2 includes a recommended task of defining how the simulation environment is "saved and restored" (IEEE Std 1730, 2011, p. 27). For sim-IS environments this task should be extended to how information is represented in ISs and the statuses of sim-IS information exchange at the time the scenario is saved, or a system error requires the sim-IS environment be restarted.

Recommended action(s). Define how sim-IS environment save and restore approach will ensure synchronization of ISs with simulations in case of a scenario pause or sim-IS environment restart. This should include identifying how IS information will be saved and reloaded. It should also include identifying how to handle the information that has been extracted from simulations but has not yet been loaded into the IS(s). If the RPA-based approach is employed for sim-IS information exchange, the RPA-based sim-IS information exchange architecture presented in Chapter IV may support restarting the RPA modules without loss of sim-IS information exchange information, as all requisite information is saved in the sim-IS data transformation components.

(6) Issue 4.2.6: Identify the Requisite Number of RPA Bot Instances for Sim-IS Information Exchange

Description. Multiple RPA bot instances may be required to meet the requirements for timeliness of sim-IS information exchange and simulation of temporal dynamics. The number of RPA bot instances may be estimated with consideration of the specified temporal dynamics (e.g., latency distribution, timeliness distribution, information submission deadline(s)), the number of simulated objects or information items for which sim-IS information exchange must be conducted, and the timeliness of the RPA modules themselves in the sim-IS environment.

Recommended action(s). Identify the requisite number of RPA bot instances for supporting sim-IS information exchange in accordance with the requirements specified for the sim-IS environment scenario. While scenario information and system documentation for the RPA modules may provide some insight into the timeliness of the RPA modules, the sim-IS environment developer should test their performance in context with hardware being leveraged for the sim-IS environment to ensure the estimated speed of the RPA modules are appropriate for the hardware and software being used for the specific sim-IS environment. This should also include identification of the maximum number of information items (in terms of units, entities, or other specific information items) to be supported by each RPA bot instance.

c. Activity 4.3: Implement Member Application Designs

> The purpose of this activity is to implement whatever modifications are necessary to the member applications so that they can represent assigned objects and associated behaviors as described in the conceptual model (Step 2), produce and exchange data with other member applications as defined by the SDEM, and abide by the established simulation environment agreements. This may require internal modifications to the member application to support assigned domain elements, or it may require modifications or extensions to the member application's external interface to support new data structures or services that were not supported in the past. In some cases, it may even be necessary to develop a whole new interface for the member application, depending on the content of the SDEM and simulation environment agreements. In this situation, the member application representative must consider both the resource (e.g., time, cost) constraints of the immediate application as well as longer-term reuse issues in deciding the best overall strategy for completing the interface. In situations where entirely new member applications are needed, the implementation of the member application design must take place at this time. (IEEE Std 1730, 2011, p. 27)

(1) Issue 4.3.1: Implement Development of IS Databases and RPA Data Transformation Component Files

Description. Having mapped the IS database(s), simulation database(s), and requisite RPA data transformation component files (if an RPA-based sim-IS information exchange approach is selected), the sim-IS developer must build the databases and data transformation components.

Recommended action(s). Develop and populate the IS databases, simulation databases, and RPA data transformation component files (if RPA-based sim-IS information exchange approach is selected) to support sim-IS information exchange in accordance with the design of the selected sim-IS information exchange mechanism(s). This should also include the division of sim-IS information exchange units or entities across sets of data transformation component files for multiple RPA bot instances, based on the estimated RPA bot instance capacities specified in activity 4.2 (Issue 4.2.5).

d. Activity 4.4: Implement Simulation Environment Infrastructure

> The purpose of this activity is to implement, configure, and initialize the infrastructure necessary to support the simulation environment and verify that it can support the execution and intercommunication of all member applications. This involves the implementation of the network design [e.g., wide area networks (WANs), local area networks (LANs)]; the initialization and configuration of the network elements (e.g., routers, bridges); and the installation and configuration of supporting software on all computer systems. This also involves whatever facility preparation is necessary to support integration and test activities. (IEEE Std 1730, 2011, p. 28)

No additional considerations are required for extending this activity to support RPA-based sim-IS environments.

5. Step 5: Integrate and Test Simulation Environment

> The purpose of this step of the DSEEP is to plan the execution of the simulation, establish all required interconnectivity between member applications, and test the simulation environment prior to execution. (IEEE Std 1730, 2011, p. 29)

Figure 62 illustrates the activities associated with Step 5 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

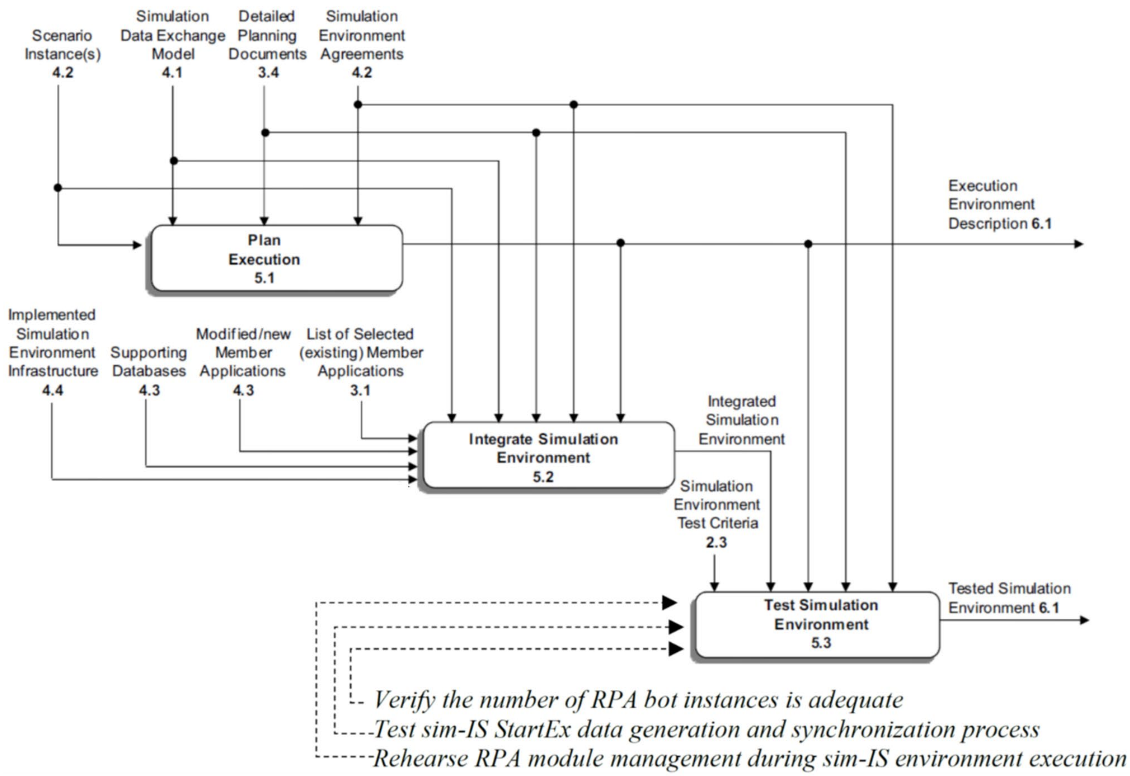


Figure 62. DSEEP Step 5: Integrate and Test Simulation Environment. Source: IEEE Std 1730 (2011) © 2011 IEEE.

a. Activity 5.1: Plan Execution

The main purpose of this activity is to fully describe the execution environment and develop an execution plan. For instance, performance requirements for individual member applications and for the larger simulation environment along with salient characteristics of host computers, operating systems, and networks that will be used in the simulation environment should all be documented at this time. The completed set of information, taken together with the SDEM and simulation environment agreements, provides the necessary foundation to transition into the integration and testing phase of development.

Additional activities in this step include the incorporation of any necessary refinements to test and VV&A plans and (for secure environments) the

> development of a security test and evaluation plan. This latter activity requires reviewing and verifying the security work accomplished thus far in the simulation environment development and finalizing the technical details of security design, such as information downgrading rules, formalized practices, etc. This plan represents an important element of the necessary documentation set for the simulation environment.
>
> Operational planning is also a key aspect of this activity. This planning should address which personnel will be operating the member applications (operational personnel) or supporting the simulation execution (support personnel) in other ways (e.g., monitoring, data logging). It should detail the schedule for both the execution runs and the necessary preparation prior to each run. Training and rehearsal for support and operational personnel should be addressed as necessary. Specific procedures for starting, conducting, and terminating each execution run should be documented. (IEEE Std 1730, 2011, p. 30)

No additional considerations are required for extending this activity to support RPA-based sim-IS environments.

b. *Activity 5.2: Integrate Simulation Environment*

> The purpose of this activity is to bring all of the member applications into a unifying operating environment. This requires that all hardware and software assets are properly installed and interconnected in a configuration that can support the SDEM and simulation environment agreements. The simulation environment development and execution plan, which is a component of the detailed planning documents, specifies the integration methodology used in this activity, and the scenario instance provides the necessary context for integration activities.
>
> Integration activities are normally performed in close coordination with testing activities. Iterative "test-fix-test" approaches are used quite extensively in practical applications and have been shown to be quite effective. (IEEE Std 1730, 2011, p. 31)

No additional considerations are required for extending this activity to support RPA-based sim-IS environments.

c. *Activity 5.3: Test Simulation Environment*

> The purpose of this activity is to test that all of the member applications can interoperate to the degree required to achieve core objectives. Three levels of testing are defined for simulation applications as follows:

> Member application testing: In this activity, each member application is tested to confirm that the member application software correctly implements its role in the simulation environment as documented in the SDEM, execution environment description, and any other operating agreements.
>
> Integration testing: In this activity, the simulation environment is tested as an integrated whole to verify a basic level of interoperability. This testing primarily includes observing the ability of the member applications to interact correctly with the supporting infrastructure and to communicate with other member applications as described by the SDEM.
>
> Interoperability testing: In this activity, the ability of the simulation environment to interoperate to the degree necessary to achieve identified objectives is tested. This includes observing the ability of member applications to interact according to the defined scenario and to the level of fidelity required for the application. This activity also includes security certification testing if required for the application. The results from interoperability testing may contribute to V&V of the simulation environment as required. (IEEE Std 1730, 2011, p. 32)

(1) Issue 5.3.1: Verify that the Number of RPA Bot Instances is Adequate

Description. The number of RPA bot instances required for supporting sim-IS information exchange depends on numerous factors, as discussed in Issue 4.2.5. During sim-IS environment testing, the performance of RPA bots in meeting the performance requirements can be tested to ensure the appropriate number of RPA bots are leveraged to achieve adequate simulation of STS dynamics through RPA-based sim-IS information exchange.

Recommended action(s). Test the performance of the RPA-based sim-IS information exchange to ensure the adequate number of RPA bot instances are employed with an adequate division of sim-IS information exchange requirements across the RPA bot instances. Testing the performance of RPA-based sim-IS information exchange performance requires the use of the intended sim-IS environment scenario and division of sim-IS information exchange responsibilities across RPA bot instances as specified in Issue 4.3.1. Limitations in RPA module timeliness may be addressed through the addition of more RPA bot instances or modifying the division of sim-IS information exchange requirements across the RPA bot instances.

(2) Issue 5.3.2: Test Sim-IS Environment StartEx Data Generation and Synchronization Process

Description. Many different elements of scenarios can be changed immediately prior to execution of a sim-IS environment. When this happens, it is the responsibility of the simulations staff to ensure all member applications and supporting infrastructure are modified and synchronized appropriately to ensure there are no negative effects on sim-IS interoperability. Thorough testing of the sim-IS environment StartEx data generation and synchronization process increases the confidence that sim-IS environment scenario changes can be implemented without negatively impacting the sim-IS integration, despite limited testing opportunities which may be available at that time.

Recommended action(s). Test the sim-IS environment StartEx data generation and synchronization process. This test should identify whether the process addresses all requisite modifications for simulation databases, IS databases, sim-IS information exchange mechanisms, and any other StartEx data files to ensure any requisite changes can be applied to the sim-IS environment scenario without degrading sim-IS interoperability.

(3) Issue 5.3.3: Rehearse Management of RPA Modules and Access to Data Transformation Component Files During Sim-IS Environment Execution

Description. The implementation of an RPA-based sim-IS information exchange approach presents multiple new exercise control processes to be examined during the testing of the sim-IS environment. This can include corrective actions, including processes required for restarting the RPA modules in case of a problem with the network or one of the member applications. It can also include routine processes for accessing the sim-IS data transformation component files to inform OTCs of any STS dynamics being simulated so the OTCs can observe how the training audience reacts to temporal or data content degradation issues.

Recommended action(s). Where an RPA-based sim-IS information exchange mechanism is employed, all processes related to the management RPA modules and use of sim-IS information exchange information for sim-IS environment control should be tested. This should include intentionally imposing errors (if possible), like a network outage, to test the process for restarting the RPA modules. The process for accessing RPA data transformation component files should be tested to ensure it does not conflict with the actions taken by the RPA modules themselves, and that it facilitates the provision of the requisite sim-IS environment information to OTCs during the exercise of the sim-IS environment.

6. Step 6: Execute Simulation

> The purpose of this step of the DSEEP is to execute the integrated set of member applications (i.e., the "simulation") and to preprocess the resulting output data. (IEEE Std 1730, 2011, p. 33)

Figure 63 illustrates the activities associated with Step 6 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

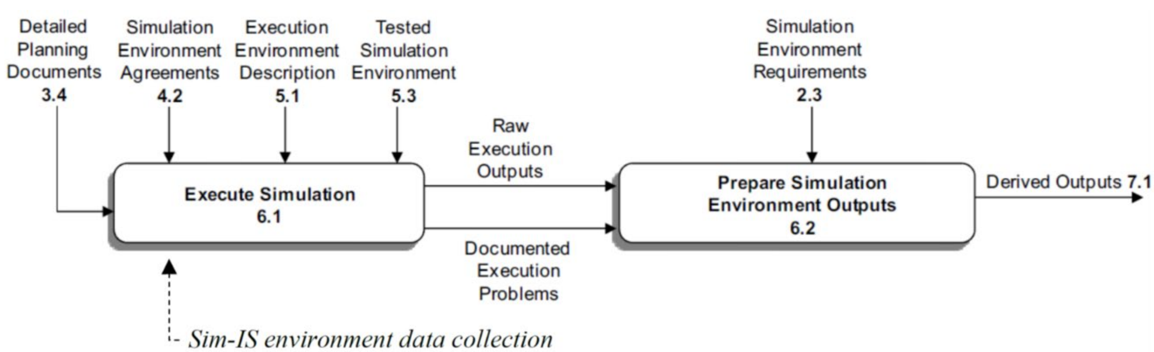


Figure 63. DSEEP Step 6: Execute Simulation. Source: IEEE Std 1730 (2011) © 2011 IEEE.

a. Activity 6.1: Execute Simulation

> The purpose of this activity is to exercise all member applications of the simulation environment in a coordinated fashion over time to generate required outputs, and thus achieve stated objectives. The simulation environment must have been tested successfully before this activity can begin.
>
> Execution management and data collection are critical to a successful simulation execution. Execution management involves controlling and

> monitoring the execution via specialized software tools (as appropriate). Execution can be monitored at the hardware level (e.g., CPU usage, network load), and/or software operations can be monitored for individual member applications or across the full simulation environment. During execution, key simulation environment test criteria should be monitored to provide an immediate evaluation of the successful execution of the simulation.
>
> Data collection is focused on assembling the desired set of outputs and on collecting whatever additional supporting data is required to assess the validity of the execution. In some cases, data is also collected to support replays of the execution (i.e., "playbacks"). Essential runtime data may be collected via databases in the member applications themselves, or can be collected via specialized data collection tools directly interfaced to the simulation environment infrastructure. The particular strategy for data collection in any particular simulation environment is entirely at the discretion of the development/integration team, and should have been documented in the simulation environment requirements, the detailed planning documents, and in the simulation environment agreements.
>
> When security restrictions apply, strict attention must be given to maintaining the security posture of the simulation environment during execution. A clear concept of operations, properly applied security measures, and strict configuration management will all facilitate this process. It is important to remember that authorization to operate is usually granted for a specific configuration of member applications. Any change to the member applications or composition of the simulation environment will certainly require a security review and may require some or all of the security certification tests to be redone. (IEEE Std 1730, 2011, p. 34)

(1) Issue 6.1.1: Sim-IS Environment Data Collection

Description. In the DMAO, Issue 6.1.1.2 identifies additional challenges associated with collecting simulation environment data from multi-architecture environments for trainee evaluations and after action reviews (IEEE Std 1730.1, 2013, p. 64). The DMAO identifies the limitation of some data collection tools for only supporting collection from certain simulation interoperability protocols as one of the complicating factors. Another is the additional complexity associated with finding the appropriate places in the multi-architecture environment to extract information to inform AARs and evaluations. Both of these issues can be even more complex in sim-IS environments.

One of the challenges for sim-IS environment data collection can be the absence of standardized protocol which can support data collection during execution of a sim-IS environment. Another issue may be the actions taken by the sim-IS environment participants interacting directly with the operational ISs. One of the items of interest in the evaluation of sim-IS environment participant performance (or JCS performance) may be identifying how the information presented in different operational ISs diverges from the ground truth information, due to limitations in the processes, user competencies and/or ISs themselves. To inform such analysis, the sim-IS environment data collection must support identification of not just the information in operational ISs relative to ground truth simulation environment information, but also identification of how the alignment (or misalignment) of said information was affected by both trainee actions and the sim-IS information exchange mechanisms employed to simulate how information is presented to sim-IS environment participants.

Recommended action(s). Test the sim-IS environment data collection plan to ensure it supports distinguishing between how sim-IS environment participant actions influence the information presented in operational ISs compared to the effects of sim-IS information exchange mechanisms simulating external STS dynamics. Where RPA-based sim-IS information exchange is used, this may necessitate modification of RPA modules (or creation of additional RPA modules) to support documentation of ground truth simulation environment information, the effects of sim-IS information exchange mechanism(s) in the presentation of that information to trainees, and the actions taken by the trainees on that information across multiple operational ISs and other information management or presentation mediums.

b. Activity 6.2: Prepare Simulation Environment Outputs

> The purpose of this activity is to preprocess the output collected during the execution, in accordance with the specified requirements, prior to the formal analysis of the data in Step 7. This may involve the use of data reduction techniques to reduce the quantity of data to be analyzed and to transform the data to a particular format. Where data has been acquired from many sources, data fusion techniques may have to be employed. The data should be reviewed and appropriate action taken where missing or erroneous data is suspected. This may require further executions to be conducted. (IEEE Std 1730, 2011, p. 35)

No additional considerations are required for extending this activity to support RPA-based sim-IS environments.

7. Step 7: Analyze Data and Evaluate Results

> The purpose of this step of the DSEEP is to analyze and evaluate the data acquired during the execution of the simulation environment (Step 6), and to report the results back to the user/sponsor. This evaluation is necessary to confirm that the simulation environment fully satisfies the requirements of the user/sponsor. The results are fed back to the user/sponsor so that they can decide if the original objectives have been met or if further work is required. In the latter case, it will be necessary to repeat some of the DSEEP steps again with modifications to the appropriate products. (IEEE Std 1730, 2011, p. 36)

Figure 64 illustrates the activities associated with Step 7 of DSEEP. Added in italics, with dashed lines, are activity tasks which have been identified for consideration in the specified activity when applying the DSEEP in support of RPA-based sim-IS environments.

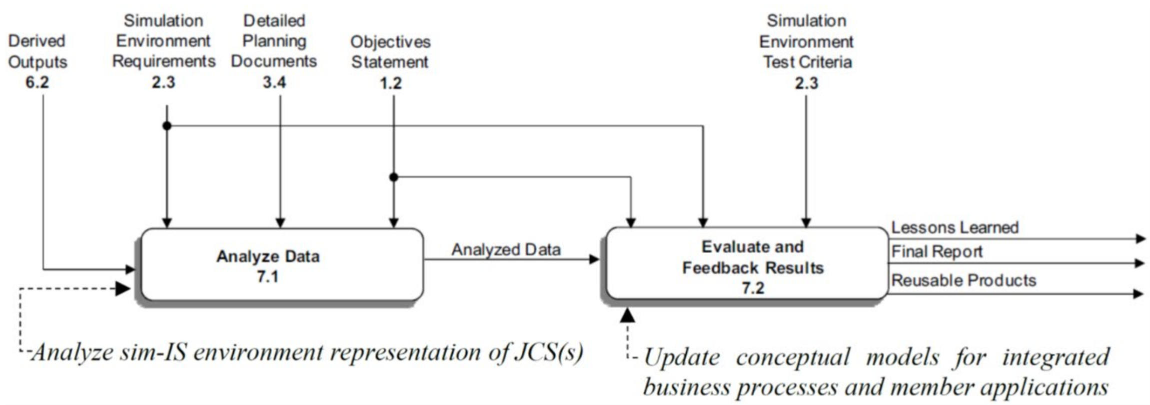


Figure 64. DSEEP Step 7: Analyze Data and Evaluate Results. Source: IEEE Std 1730 (2011) © 2011 IEEE.

a. Activity 7.1: Analyze Data

> The main purpose of this activity is to analyze the execution data from Step 6. This data may be supplied using a range of different media (e.g., digital, video, audio), and appropriate tools and methods will be required for analyzing the data. These may be commercial or government off-the-shelf (COTS/GOTS) tools or specialized tools developed for a specific simulation environment. The analysis methods used will be specific to a particular simulation environment and can vary between simple observations (e.g., determining how many

> targets have been hit) to the use of complex algorithms (e.g., regression analysis or data mining). In addition to data analysis tasks, this activity also includes defining appropriate formats for presenting results to the user/sponsor. (IEEE Std 1730, 2011, p. 36)

(1) Issue 7.1.1: Analyzing Sim-IS Environment Representation of JCS

Description. Sim-IS environments afford an opportunity to observe performance of some of the integrated business processes that they are simulating. Depending on the generalizability of the scenario and sim-IS environment participants, comparison of the simulated integrated business processes and associated STS dynamics may be valuable for informing the continuous refinement of the sim-IS environment and the sim-IS information exchange mechanism(s) and simulated STS dynamics in particular.

Recommended action(s). Identify how integrated business processes observed during execution of the sim-IS environment may inform the design of the sim-IS environment itself. Analyze the differences between observed integrated business processes and associated STS dynamics and those simulated by the sim-IS environment, and sim-IS information exchange mechanisms in particular. It may be useful to maintain a library of conceptual models for the different processes and STS dynamics observed for different sets of sim-IS environment participants, with documentation of the relevant scenario information and participants (e.g., the participants' stage in the forming of their team, levels of experience among different trainee staff members, trainees' understanding of their organizational processes). This analysis is important for informing the design of the sim-IS environment as well as providing feedback for the coevolution of the exercised JCSs themselves.

b. Activity 7.2: Evaluate and Feedback Results

> There are two main evaluation tasks. In the first task, the derived results from the previous activity are evaluated to determine if all objectives have been met. This requires a retracing of execution results to the measurable set of requirements originally generated during conceptual analysis (Step 2) and refined in subsequent DSEEP steps. This step also includes evaluating the results against the test and execution "pass/fail" criteria for the simulation environment. In the vast majority of cases, any impediments to fully satisfying the requirements have already been identified and resolved during the earlier development and integration phases. Thus, for well-designed simulation

> environments, this task is merely a final check. Having completed this first evaluation process, the conclusions should be communicated to the user/sponsor. In those rare cases in which certain objectives have not been fully met at this late stage of the overall process, then, with the user/sponsor's approval as there may be cost and time implications, corrective actions should be identified and implemented. This may necessitate revisiting previous steps of the DSEEP and regenerating results.
>
> The second evaluation task in this activity is to assess all products generated in terms of their reuse potential within the domain or broader user community. Those products identified as having such reuse potential should be stored in an appropriate archive. Examples of potentially reusable products include the scenario and the conceptual model, although there may be several others. In fact, it may be advantageous in some instances to capture the full set of products required to reproduce the execution of the simulation environment. Determination of which products have potential for reuse in future applications is at the discretion of the development/integration team. (IEEE Std 1730, 2011, p. 37)

(1) Issue 7.2.1: Update Conceptual Models for Target Processes and Member Applications Based on Lessons Learned in Observing JCSs in Context

Description. RPA conceptual models may be even more critical for the design, development, and management of RPA workflows supporting RPA-based sim-IS environments as those supporting operational work environments. In sim-IS environments, RPA must adapt to not only changes in business processes and UIs, but also changes in the simulation systems employed to simulate the integrated business processes as simulations change over time as well. RPA module conceptual models should be continuously updated to reflect changes in simulations, operational ISs, and the latest understanding of the JCS to be simulated.

The DSEEP describes Activity 7.2 as a "final check" of the simulation environment to ensure it meets the sponsor's requirements for simulation of the referent (IEEE Std 1730, 2011, p. 37). For sim-IS environments simulating JCSs, particularly those JCSs which are not readily accessible for observation in the real-world (e.g., in combat operations centers or emergency management operations), an additional perspective should be applied. Activity 7.2 affords an opportunity to continuously refine the understanding and representation of the JCS(s) as they continuously coevolve. The analysis of divergence between observed integrated business processes and STS dynamics and the simulated processes and STS dynamics (identified in Activity 7.1) should be evaluated to determine the extent to which any divergence in anticipated processes and STS dynamics represents a change which is generalizable across the enterprise. If a divergence in processes or STS dynamics is identified, the evaluation conducted in Activity 7.2 should determine whether it represents a change in enterprise processes or an example of variability in the integrated business process(es) across the enterprise. In either case, updated conceptual models may be required for the target integrated business processes, the sim-IS environment, and the member applications as changes are mirrored across all three levels.

Activity 7.1 is also described as providing an opportunity for evaluating the potential for simulation environment components for reuse. For simulation of JCSs and associated STS dynamics, this should be viewed as linked to the previous issue of refining the sim-IS environment's representation of the target process(es) and associated STS dynamics. By maintaining updated conceptual models for the target environment, the sim-IS environment, and the member applications would facilitate the effective and efficient evaluation of the sim-IS environment and its member applications for future reuse as JCSs continuous evolve.

Recommended action(s). Update conceptual models to reflect any lessons learned regarding the simulated JCSs and associated STS dynamics. After analyzing observations from sim-IS environment execution, determining the generalizability of the observations and the requisite changes for how target integrated business processes and STS dynamics should be simulated, the sim-IS designer should ensure all affected conceptual models are updated for future use. By maintaining synchronized conceptual models for the target phenomenon (or processes), sim-IS environment, and sim-IS environment member applications (including sim-IS information exchange mechanism(s)), the sim-IS environment designer can support the continuous coevolution of the JCS(s) which the sim-IS environment is intended to support.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSIONS

With the proliferation of increasingly powerful information systems in naturalistic decision-making environments, sim-IS environments are needed which enable military services to exercise integrated business processes and develop the intuition of commanders and staffs in context. Conventional sim-IS information exchange approaches are limited in their ability to integrate simulations and HitL ISs and to simulate the STS dynamics associated with the simulated integrated business processes. This research examined the potential for an RPA-based outside-in approach to sim-IS information exchange to overcome these limitations.

Two design science research artifacts were developed in support of this research: a prototype RPA-based sim-IS information exchange architecture (*instantiation artifact*) and recommendations for a DSEEP overlay for RPA-based sim-IS environments *(method artifact*). M&S verification and validation of the instantiation artifact supported validation (in the DSR sense of the word, as discussed in Chapter III) of the RPA-based approach to sim-IS information exchange and simulation of integrated business processes.

This research contributes to filling a gap in the simulation of integrated business processes in sim-IS environments for staff training. The prototype RPA-based sim-IS information exchange architecture presents a new approach for integrating simulations and HitL ISs through RPA technology. The recommendations presented for a DSEEP overlay for sim-IS environments provide guidance for the design, development, and reuse of RPA-based sim-IS environments. This research constitutes first steps in the exploration of an outside-in approach to the design and development of RPA-based sim-IS environments. Additional research is warranted for enhancing the simulation of integrated business processes and STS dynamics. Additional research is also necessary for exploring the potential application of RPA-based sim-IS environments to support emergency management training and simulation of cyber-effects in staff training.

A. RESEARCH CONTRIBUTIONS

This research established the potential for an RPA-based approach to sim-IS information exchange to support the simulation of integrated business processes and associated STS dynamics for staff training. This new sim-IS integration approach presents opportunities for the inclusion of HitL ISs in simulation-supported staff training environments, the simulation of integrated business processes, and the representation of STS dynamics in how information is presented to sim-IS environment participants. Research contributions include:

1) validation of the proposed RPA-based outside-in approach to sim-IS information exchange as a new means of automating sim-IS information exchange (research question 2),
2) validation of the proposed RPA-based approach to sim-IS information exchange for simulating integrated business processes and associated sociotechnical system dynamics (research question 1),
3) a prototype modular architecture for RPA-based sim-IS environments and technique for estimating the requisite number of RPA bots for a given sim-IS environment (research questions 1, 2 and 3), and
4) recommendations for a DSEEP overlay to guide the engineering and execution of sim-IS environments with an emphasis on RPA-based sim-IS information exchange considerations (research questions 3 and 4).

1. Validation of the RPA-based Sim-IS Information Exchange Approach

This research validated (in the design science research sense of the word) the RPA-based approach to sim-IS information exchange as a new means of automating sim-IS information exchange and simulating integrated business processes and associated STS dynamics. This was achieved through the verification and validation (in the M&S community sense of the word) of the prototype RPA-based sim-IS information exchange architecture.

Verification of the prototype RPA-based sim-IS information exchange architecture for sim-IS information exchange and simulation of STS dynamics was conducted through testing of the architecture in two sim-IS environments (MTWS-CLC2S and JCATS-CLC2S). Statistical significance was observed in the execution of sim-IS information exchange in compliance with target performance thresholds and in the simulation of temporal dynamics in the form of information exchange latency and timeliness variability. Practical significance, but not statistical significance, was observed in the simulation of specified information content degradation probabilities. These results establish the potential for an RPA-based approach to sim-IS information exchange to support the simulation of STS dynamics in sim-IS environments.

Validation of the prototype architecture for supporting sim-IS environments in a staff training context was conducted through solicitation of SME feedback following demonstration of the prototype aboard MCAGCC. The MCLOG M&S section personnel, who also constitute this work's research sponsors, found that the proposed RPA-based sim-IS information exchange architecture would yield sim-IS environments with greater representation of operational ISs and simulation of STS dynamics than are feasible with conventional sim-IS interoperability methods. This, combined with the quantitative analysis results from the verification process, provides evidence of the architecture's capacity for enhancing sim-IS environments for staff training.

As discussed in Chapter VI, validation of the RPA-based sim-IS environment is necessarily a continuous process, owing to the continuous coevolution of joint cognitive systems and the need for associated RPA-based sim-IS environments to coevolve with the JCSs they are designed to simulate. Continuous validation of RPA-based sim-IS environments requires processes for the continuous evaluation and refinement of conceptual models for the simulated JCS and RPA-based sim-IS environments simulating the JCS. This issue is one of many addressed in the recommendations provided in Chapter VII for a DSEEP overlay for sim-IS environments.

2. Prototype RPA-based Sim-IS Information Exchange Architecture

The RPA-based sim-IS information exchange architecture itself provides a first step in the design of an RPA-based approach for sim-IS information exchange. The architecture was designed in a modular fashion to leverage the existing strengths of RPA platforms for workflow reuse. The architecture presented in this research is modular in the sense of coupling RPA modules for different simulations and ISs, as well as for spreading sim-IS information exchange requirements across multiple RPA bot instances.

In addition to establishing the potential of an RPA-based approach to sim-IS information exchange, this research identified approaches for overcoming various issues identified as needing to be addressed for the design and development of RPA-based sim-IS environments. One such issue is determining the number of RPA bot instances necessary for supporting a particular sim-IS environment. An approach for estimating the number of RPA bot instances necessary for a given set of information items and timeliness distributions (and the RPA bot performance data which must be measured to support this approach) is specified in Chapter V. Where and how to apply these calculations in the broader process for the engineering and execution of RPA-based sim-IS environments is specified in DSEEP overlay recommendations provided in Chapter VII.

Another important issue that must be resolved to implement an RPA-based sim-IS information exchange approach is the alignment of sim-IS parametric data and the generation of StartEx data for simulations, ISs, and RPA workflows. Building on the existing OBS standard, a practical approach is presented for automating the generation and synchronization of StartEx data for simulations, logistics ISs, and RPA modules for RPA-based sim-IS environments. While the structure of the RPA data transformation components and BFG application for addressing these issues are specified in Chapter IV, guidance for the generation, maintenance, and reuse of products in the context of the DSEEP are documented in the recommendations provided for a DSEEP overlay for RPA-based sim-IS environments.

3. DSEEP Overlay Recommendations for Sim-IS Environments

Simulation environment developers must assume responsibility for StartEx data synchronization with any operational information systems to be stimulated by their simulations, whether the dynamic sim-IS information exchange is conducted by pucksters or automated RPA bots. If they do not, the misalignment of simulations and operational ISs can derail even the most carefully planned exercise. With the growing number and importance of operational ISs for commanders and staff processes, sim-IS environment developers face an overwhelming challenge in the design and development of sim-IS environments for staff training.

Unfortunately, existing simulation environment tools (e.g., JTDS, OBS standard) and guidelines (e.g., DSEEP) provide little to no support for the engineering and execution of sim-IS environments, focusing instead on the integration of distributed simulations only. This research addresses this gap by presenting tools and techniques to support an automated approach to the generation and synchronization of sim-IS StartEx data and recommendations for a DSEEP overlay to guide the engineering and execution of sim-IS environments. In addition to addressing general considerations for the alignment of simulations and ISs in sim-IS environments, the DSEEP overlay recommendations address specific issues related to the engineering and execution of RPA-based sim-IS environments. These recommendations are informed by the iterative design, development, verification, and validation of the RPA-based sim-IS information exchange architecture and prototype RPA modules presented in this research. As such, as discussed in the previous section, many of the issues which are identified directly relate to design considerations in the RPA-based sim-IS information exchange architecture presented in Chapter IV.

The DSEEP overlay recommendations presented in this research also serve to facilitate the design and development of sim-IS environments for other purposes beyond DOD staff training environments. While this research discusses the DSEEP in the context of supporting the design and development of sim-IS environments for training, the DSEEP also supports the engineering and execution of distributed simulation environments for other purposes, such as analysis, wargaming, and experimentation. The low-overhead, modular nature of RPA-based sim-IS information exchange also makes this proposed DSEEP overlay a potentially useful guide for the use of RPA-based sim-IS environments outside the DOD, where fewer resources may be available for the engineering and execution of sim-IS environments. The next section addresses future research opportunities for employment of the proposed RPA-based sim-IS environments in other domains as well as additional research for enhancing staff training environments.

B. FUTURE WORK

This research has demonstrated the potential for RPA technology to support a new means for automating sim-IS information exchange and simulating integrated business processes and associated STS dynamics. Additional research is necessary, however, to better understand how to design, develop, and maintain an RPA-based approach to sim-IS information exchange.

> A major finding of MSG-086 (besides the detailed documentation of simulation interoperability issues) is that simulation interoperability is not primarily a technical issue, but that simulation interoperability needs to be addressed in a holistic way along the whole simulation engineering process. (NATO Science and Technology Organization, 2015a, p. 1)

Issues requiring future research for the RPA-based sim-IS environments include a blend of technical and non-technical considerations. This is not surprising, as some of the greatest obstacles for simulation interoperability are non-technical, as reported in the final report of NATO task group MSG-086 in 2015; also see (Blais, 2022). An example of such a non-technical problem for U.S. Marine Corps simulations is the misalignment of simulation visual models and parametric data (including enumerations), which presents an obstacle for on-demand integration of LVC simulations (Training and Education Capabilities Division, Training and Education Command, USMC, 2018).

The simulation of integrated business processes which coevolve with the RPA-based sim-IS environments themselves requires the development of conceptual modeling tools and practices to support the modeling of the integrated business process, STS dynamics, and RPA workflows. It also requires the continued development of standards which guide the development of sim-IS environments. While the previous section identified the need for continued development of a DSEEP overlay for sim-IS environments, additional research should also be applied to extending the Standard for Federation Engineering Agreements Template (SISO-STD-012-2013) to support the reuse of RPA-based sim-IS environment components. There are also opportunities to explore regarding the application of an RPA-based sim-IS information exchange approach to support simulation in other communities (e.g., emergency management operations) and other warfighting domains (e.g., cyber).

1. Extending FEAT for Sim-IS Environments

The proposed DSEEP overlay for RPA-based sim-IS environments is intended to provide guidance for the engineering and execution of sim-IS environments, including the use of RPA-based sim-IS information exchange. An additional standard which may benefit from such an overlay for supporting RPA-based sim-IS environments is the FEAT (SISO-STD-012-2013).

> The Federation Engineering Agreements Template (FEAT) provides a standardized format for recording federation agreements to increase their usability and reuse. The template is an eXtensible Markup Language (XML) schema from which compliant XML-based federation agreement documents can be created…The FEAT benefits all developers, managers, and users of distributed simulations by providing an unambiguous format for recording agreements about the design and use of the distributed simulation. (SISO FEAT Product Support Group, 2017, p. 6)

The FEAT consists of eight categories, shown in Figure 65, and provides standardized formats to document relevant information about distributed simulation environments across subsections within each category. This documentation supports reuse of distributed simulation environments, or components of the environments, in support of future simulation requirements. A FEAT overlay for sim-IS environments could support the documentation of sim-IS environments to support the reuse of sim-IS information exchange mechanisms. Such documentation of sim-IS environments could facilitate more efficient or effective resolution of issues at various levels of interoperability, from technical to conceptual, in the design and development of future sim-IS environments.

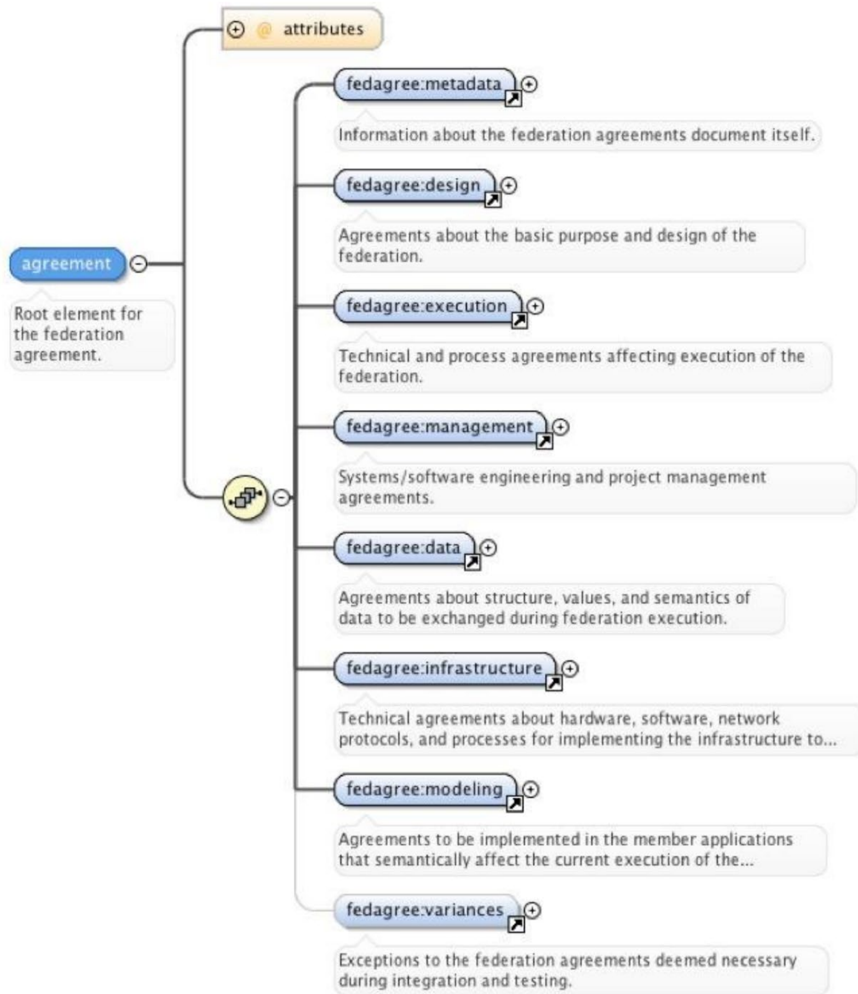


Figure 65. Top-level View of the FEAT Schema. Source: SISO FEAT Product Support Group (2017).

At the conceptual level, for example, FEAT documentation of conceptual models for the sim-IS environment and member applications could support analysis of their appropriateness for supporting the simulation of integrated business processes in the future. This could be addressed in the conceptual model subsection of the FEAT's design agreements category, with the documentation identifying the relevant conceptual models for different member applications, how different components of the respective conceptual models relate to components of the sim-IS environment conceptual model, the degrees with which they align and any notable limitations. This could even include recommendations to the original FEAT standard to support the documentation of conceptual models for member applications, as identified for DSEEP.

An adjusted FEAT may also support documentation of sim-IS environments at the technical, semantic, and syntactic levels of interoperability. Documentation of numerous issues addressed in the DSEEP overlay recommendations provided in Chapter VII could be supported in the data agreements category (e.g., data exchange models and supporting databases subsections) and the infrastructure agreements category (e.g., middleware, secondary communication channels, and hardware configuration subsections). Future research is needed to identify how the FEAT standard may be adjusted or extended to support documentation of sim-IS environment conceptual models as well as unique sim-IS information exchange considerations ranging from StartEx data generation and synchronization to the design of RPA-based sim-IS information exchange modules.

2. Military and RPA Domain Ontologies

Conceptual modeling of RPA-based sim-IS environments for military integrated business processes depends on future work in the development of domain ontologies and practices for integrating conceptual models developed using these domain ontologies. Future work is needed to develop domain ontologies for the representation of platform-independent RPA workflows and the representation of integrated business processes, and associated STS dynamics, in the military domain. This would require the development of a platform-independent RPA domain ontology, building on the work of Völker and Weske (2021) discussed earlier, as well as reinvigoration of earlier conceptual modeling efforts in the DOD.

In 1997, Rybacki and Blackman (1997) identified the potential value of a logistics M&S taxonomy to "enable communication between automated information systems that are used in the performance of [logistics functions] and the models and simulations (M&S) that are used to represent those functions" (Rybacki & Blackman, 1997, p. 1). Rybacki and Blackman proposed that such a logistics M&S taxonomy would be developed as part of a broader DOD conceptual modeling effort initiated twenty-five years ago: Conceptual Models of the Mission Space (CMMS). CMMSs are defined by the DOD Modeling and

Simulation Office (DMSO, now the DOD Modeling and Simulation Enterprise) as "simulation implementation-independent functional descriptions of the real world processes, entities, environmental factors, and associated relationships and interactions constituting a particular set of missions, operations, or tasks" (Sheehan et al., 1998, p. 1). The CMMS effort was established by the DMSO in the 1995 DOD M&S Master Plan as a "common starting point and eventual real-world baseline for consistent and authoritative [M&S] representations" (Sheehan et al., 1998, p. 1). Rybacki and Blackman (1997) stated that the taxonomy they proposed for logistics and associated AISs should be maintained as a component of either the CMMS or related WarSim Functional Description of the Battlespace effort conducted under the Army as a conceptual model repository support of WarSim 2000 development. Unfortunately, the concerted CMMS effort was not maintained for long by DMSO (Liu et al., 2011). While similarities exist between the FEDEP/DSEEP conceptual modeling step and CMMSs, by 2011 no evidence remained of a continued DMSO effort for maintenance of the CMMS program for a DOD domain ontology (Liu et al., 2011).

With the modern pursuit of higher levels of interoperability, M&S-supported wargaming, and MSaaS, some in the DOD M&S community are calling for greater effort to be placed in conceptual models in support of M&S (Kackley, 2022). While renewing M&S CM efforts, special attention should be applied to the lessons learned during the previous DOD CM push, such as those presented in (Pace, 2010). This should include consideration of the shortfalls of the last CMMS effort. It should also include consideration of how a renewed DOD M&S conceptual modeling effort could support representation of the role of automated information systems in support of military operations.

3. IS to Simulation Information Exchange

This research focused on the automated exchange of information from simulations to operational ISs to inform training audience personnel about events in the simulated environment. An additional approach for RPA-based sim-IS information exchange which should be explored is the automated exchange of information from operational ISs to simulations. There are a couple types of environments which could benefit from an automated flow of information from ISs to simulations, including both training and operational wargaming environments.

Automated information exchange from ISs to simulations could be developed to support the use of operational ISs in staff training or wargaming environments as a means of directing actions of simulated entities. There are two general approaches that this could result in, based on the degrees with which simulated unit actions could be anticipated. In most instances, the support requests or orders communicated via operational ISs alone are insufficient for determining how receiving units would respond. Responding to maintenance or transportation support requests, for example, would require consideration of numerous additional factors, including the priority of the request relative to requests from other units, the availability of the requested resource(s), and the resources necessary for providing the requested support. While such requests or orders are often adjudicated by response cell personnel in staff training exercises, future research should explore the potential for so-called "smart RPA" to support the automation of some actions in response to training audience communication.

In some instances, the information entered into operational ISs may be sufficient for supporting the simulation of actions by simulated entities. An example of this is the submission of ground transportation requests (GTRs) in the TCPT system. GTRs include information regarding the composition of ground convoys (personnel, equipment, and transported supplies), the planned route, and the departure time. This information may be sufficient for automating the generation and execution of simulated ground movements in simulation systems supporting a sim-IS environment. Additional research is necessary to identify whether such a use of RPA-based sim-IS information exchange would present any additional issues requiring consideration in StartEx data generation and the design of the RPA-based sim-IS architecture.

Automated information exchange from operational ISs to simulations could also support the initialization of sim-IS environments for operational wargaming environments. Previous research in sim-C2 interoperability, including the work of NATO MSG-085, has identified the potential value of C2-to-simulation information exchange for supporting course of action analysis in an operational environment (Burland et al., 2014; *Standardisation for C2-Simulation Interoperation*, 2015).

> Beyond the domain of training, the ability to couple C2 and simulation systems presented an intriguing new possibility: that simulation could support planning and preparation phases of ongoing military operations, providing course of action analysis and mission rehearsal capabilities. (Pullen & Khimeche, 2014, p. 1)

As with training, conventional approaches for sim-C2 integration in support of course of action analysis are limited in their ability to support integration with HitL ISs. This can present a challenge for automating the modification of simulations to represent force readiness levels, as C2 systems often lack detailed readiness information such as equipment readiness, personnel casualties, and supply statuses. An RPA-based approach to IS-to-simulation information exchange presents an opportunity for addressing this limitation by supporting the automated modification of simulated entity readiness levels in accordance with readiness information found in operational logistics ISs.

4. Event-based Information Exchanges

Thus far, exploration of an RPA-based approach for sim-IS information exchange has only addressed simulation of routine, scheduled information exchange events. Scheduled information exchange requires scheduling a single data extraction event for the corresponding data entry event. Simulation of event-based information exchange events (e.g., CCIR reporting), however, would require continuous sampling of certain simulation values to determine when a reporting event, or CCIR threshold, has been triggered. This would require changes to the RPA modification module, with potential implications for the broader RPA-based sim-IS information exchange architecture.

For CCIR events, the RPA modification module must first determine whether the CCIR threshold has been reached (e.g., howitzer readiness falls below 90%). If it has, then the modification module could schedule an event for submitting a message to the requisite CCIR holder, be this through the associated IS or some other text-based communications software. While the readiness levels for a unit will be transmitted once a day via logstats, the CCIR events should trigger a faster reporting of the associated information, in both the logistics information system and other appropriate reporting systems. Future research is needed to examine how an RPA-based information exchange approach may be applied to simulate such event-based information reporting.

5. Simulation of Cyber Attacks

The simulation of information degradation resulting from sociotechnical system dynamics is closely related to another important research domain in M&S: the simulation of data manipulation resulting from cyber-attacks. Unfortunately, despite the proliferation of cyber training ranges, the representation of cyber effects is similarly lacking in battle staff training exercises across the DOD. Where cyber operators are included in broader DOD exercises, there are often gaps between the simulation environment member applications supporting traditional staff training and those supporting cyber training (Morse et al., 2014; Wells & Bryan, 2015). When cyber effects are simulated in staff training exercises, they are simulated through manual MSEL injects (i.e., white card events) or with M&S tools which are limited in the types of cyber-attacks which can be represented. Multiple research efforts have been conducted, or are currently underway, to address these gaps. The Cyber Operational Architecture Training System (COATS) was developed to improve the integration of cyber training in larger DOD exercises by "reducing or eliminating gaps and seams between traditional and cyber training architectures" (Morse et al., 2014, p. 3). The SISO Cyber M&S Study Group was established to "identify key cyber M&S activities, document best practices, highlight lessons learned, and identify areas for potential standardization in order to facilitate adoption by the cyber M&S community" (SISO Standards Activity Committee, 2020, p. 5).

Cyber-attacks can range from network-oriented attacks (e.g., denial of service) to software-oriented attacks (e.g., tampering attacks or ransomware). Cyber-attacks can also be conducted for a variety of purposes, including reconnaissance, disruption, degradation, or manipulation. The difference between manipulation and disruption or degradation is notable for this research.

> ...what is the difference between a manipulation that achieves a degradation or a direct degradation as a tactic? It is a very subtle one: mainly, manipulation refers to a manner that is not [immediately] apparent or detected: a DDoS (degradation or disruption) or a ransomware attack (destruction) are immediately identified by the victim. (Villalón-Huerta et al., 2021, p. 5)

Although manipulation is one of the more dangerous types of cyber-attacks (Mudge & Lingley, 2008), it is more difficult to simulate in sim-IS environments. Conventional approaches for simulation of cyber-attacks in staff training are generally limited in their representation. Distributed denial of service (DDoS) attacks can be simulated through complete or partial dropping of information exchange packets transmitted through communications effects servers, as discussed in Chapter II. While communications effects servers may also support some simulation of spoofing track data, and modification of other messages traversing servers during sim-IS exchange (Pullen & Ruth, 2018b), they are generally not designed for accessing ISs directly to manipulate information.

The RPA-based approach to sim-IS information exchange presents a potential new means for simulating these types of cyber effects. Furthermore, the detection of cyber-attacks for manipulation is complicated by the task of distinguishing between user data entry errors and malicious data manipulation imposed by cyber-attacks. Sim-IS environments which simulate the routine information degradation resulting from STS dynamics and information degradation resulting from cyber-attacks present an opportunity for stressing battle staff processes for detection and response to potential cyber-attacks. Future research is necessary to explore how an RPA-based approach to sim-IS information exchange can support the simulation of cyber effects, and how to couple it with the simulation of STS dynamics to stress staff processes for cyber effects detection.

6. Supporting Emergency Management Operations Training

Simulation of integrated business processes in context is important for any organization seeking to improve decision-making training and support the coevolution of joint cognitive systems. It is especially important, however, for those organizations which lack consistent opportunities to exercise and observe critical processes in context. This includes the military services as well as the many organizations responsible for emergency management operations in response to civil and natural disasters. “Apart from its obvious value in training, simulation may also be an efficient tool for [emergency] response system development, because experiences drawn from exercises may reveal unexpected demand in real-world contexts” (Bram & Vestergran, 2012, p. 43). An RPA-based approach to sim-IS information exchange presents an opportunity for the development of valuable training and JCS development environments for emergency management organizations.

Across the United States, federal, state, and local emergency responders provide daily services to their communities in such occupations as firefighters, police officers, and city planners. Throughout the year, in addition to their regular duties, these civil servants prepare to conduct emergency management operations in response to large scale emergencies, either natural or man-made, that could affect their communities. A central consideration in emergency management preparations is the establishment, manning, and operations of an emergency operations center (EOC). Like military COCs, EOCs serve as command and control node for emergency responders, coordinating actions across adjacent organizations and between levels of command. EOCs are generally established at city, county, and state levels, with FEMA fielding a federal EOC as appropriate (Pine, 2018).

EOCs often employ management information systems to assist in the management of internal and external coordination, including submission and receipt of support requests, tracking supply statuses, and identifying the location of teams engaged in the situation. While FEMA relies upon many management information systems, the Web EOC (WebEOC) IS plays a particularly vital role in supporting emergency management coordination across federal, state, and local EOCs (*Using Innovative Technology and Practices to Enhance the Culture of Preparedness*, 2018).

WebEOC is a commercial web-based IS that is used by FEMA and by state and local EOCs to manage emergency operations (Pine, 2018). Although many EOCs use WebEOC, some states and local authorities use different information management systems. Even among those who do use WebEOC, all instances of the system are not interoperable across these different levels. In a 2015 report, titled “FEMA Faces Challenges in Managing Information Technology,” the Office of the Inspector General (OIG) reported shortfalls in the interoperability of the nation’s emergency management information systems (Office of the Inspector General, 2015). In addition to identifying inadequate interoperability between the FEMA WebEOC system and other federal information management systems, the OIG found that "the FEMA WebEOC is not integrated with the WebEOC used by state emergency operation centers" (Office of the Inspector General, 2015, p. 22). The report proceeded to describe the manpower and time-intensive manual processes imposed upon emergency management personnel attempting to use WebEOC for external coordination.

> FEMA regions rely on an inefficient manual process to update the FEMA WebEOC with information from the state centers about ongoing disasters. Specifically, a region has to send FEMA staff to a state emergency operation center to review the state's information. If a state's request for assistance is submitted in the state system, a FEMA staff member must print it out and manually enter the same data into the FEMA WebEOC. This process can cause delays in providing disaster assistance. For example, during an exercise in one state, FEMA staff had to manually transfer 18 state requests from the state system into the FEMA system before FEMA could process the requests. According to FEMA staff, this caused a delay of between 2 to 6 hours, which can be critical in emergency management and response, which involves saving lives and preventing property damage. (Office of the Inspector General, 2015, p. 22)

Just as limited interoperability between local, state, and federal emergency management/WebEOC systems can degrade operational coordination, it can also make the representation of ISs for EOC training exercises prohibitively manpower intensive. Future research should explore how an RPA-based approach to sim-IS information exchange can support the stimulation of WebEOC systems in sim-IS environments. Such sim-IS environments may support the representation of WebEOC interoperability shortfalls, supporting the refinement of WebEOC information exchange processes and training of emergency management professionals manning EOCs at various levels.

7. Envisioning a Cycle of Training

In recent years, M&S technology and wargames have grown in esteem among senior military leaders. In 2014, Secretary of the Defense Chuck Hagel called for a "reinvigorated wargaming effort" across the DOD (Wong et al., 2019, p. 19). In 2019, the Commandant of the Marine Corps emphasized the value of M&S and wargames in support of his Marine Corps force redesign efforts, as well as for training and education

(Headquarters Marine Corps, 2019). In this climate, where their primary tools enjoy such trust among senior decision makers, it is essential that M&S, operations research, and wargaming professionals are particularly vigilant in ensuring limitations of their individual tools are understood by both their customers and fellow practitioners. These communities should also take advantage of the renewed interest to improve the integration of the wargaming, combat modeling, and training exercises with the objective of bolstering respective tools' strengths and mitigating weaknesses.

Analysts and wargaming professionals often refer to the "cycle of research," illustrated in Figure 66, where wargaming, combat modeling, and exercises are integrated in order to achieve more comprehensive, thorough analysis than could be achieved through the use of any one tool (Perla, 2011, p. 251). Each element of this cycle provides a valuable perspective and capability to benefit analysts and decision makers, with no one tool being sufficient to adequately inform analysis. Wargames facilitate training for decision making and support analysts as they explore new technology and concepts. Combat models support the automation of numerous wargaming and training tasks, increasing the scope and fidelity that can be achieved in training and wargaming tasks. Analysis provides insights into the relative combat power of different tactics or weapons and can identify potential threats or opportunities to wargame further. Training exercises and historical analysis provide insights into the effects of human factors on the battlefield, which can benefit both wargames and analysis.

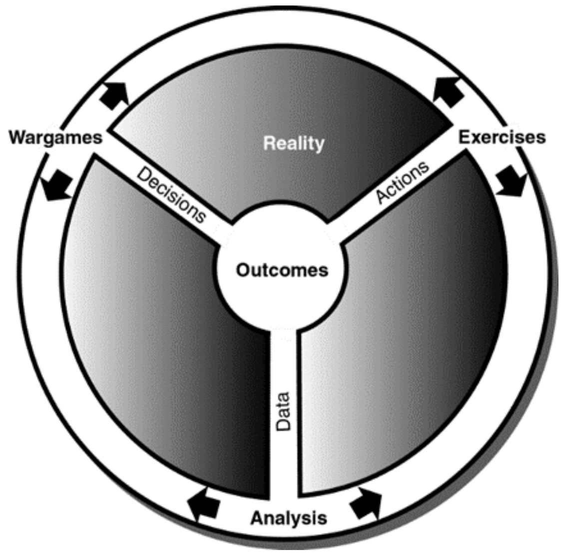


Figure 66. The Cycle of Research. Source: Perla et al. (2019).

In the mid-twentieth century, such tight integration may have seemed obvious. With the end of second world war, analysts saw wargames as tools for “developing the necessary hypotheses” regarding the impact of the new technology on the battlefield (Rowland, 2019, p. 14). However, the integration of the wargaming and combat modeling communities has degraded over the last few decades, with some going so far as to assert that the cycle of research is “broken” (McGrury, 2019). Some wargaming experts have found that combat models have lost their basis in the data and insights gleaned from real world combat and observation of field exercises (Lawrence, 2017; Rowland, 2019). The disintegration of ties across the different elements of the cycle of research also threatens to undermine the “objectivity, rigor, and usefulness” of each tool and analysis process as a whole (Hanley, 2017, p. 67).

Despite the several challenges facing those pursuing a cycle of research, renewed emphasis on wargaming presents an opportunity to overcome previous obstacles and develop new capabilities. Furthermore, while the cycle of research resonates in the tools’ support of analysis, an opportunity exists to develop a distinct “cycle of training” approach to improve the way sim-IS environments for training and wargaming support the coevolution of joint cognitive systems. RPA-based sim-IS environments present an opportunity to develop wargames that better simulate decision-making environments.

> The cycle of research for war gaming and combat/campaign simulation also extends to studying history and developments in social science, including experimental gaming on human behavior (such as in behavioral economics) and cognitive science studying developments in understanding the brain, etc., to explore human reasoning and dynamics. (Hanley, 2017, p. 96)

Despite the intended focus of wargaming on decision-making, cognitive science, decision-making theories, and cognitive task analysis techniques are rarely addressed in the design and development of wargames. While advances in such domains as naturalistic decision-making and situated cognition have had considerable impact on the design of training environments, they are rarely mentioned in the design and development of wargames. As the Marine Corps pursues "an 'unprecedented' level of immersion with its wargames, moving away from the wargaming's dominant board game-style format" (Wong et al., 2019, p. 32), Marine Corps wargaming capability developers should also consider how they can simulate the processes and artifacts through which information about the battlefield is delivered to the decision-makers and their staffs. Here the distinction between ground truth and perceived truth is as important for the design of wargaming environments as it is for the design of sim-IS environments for training. Kackley (2022) reports that conceptual models are being developed in support of the Marine Corps Wargaming and Analysis Center. As with simulation-supported training environments, by opening the aperture for what is included in these CMs to include the integrated business processes through which information is presented to decision-makers, the wargaming center can design wargaming environments which support the exercising and observation of decision-making in context.

As the DOD pursues advanced AI-enabled decision-support systems, the importance of representing ISs in the context of associated integrated business processes for wargaming environments should be expected to grow. The RPA-based approach to sim-IS information exchange presents an opportunity for achieving sim-IS wargaming environments which simulate the integrated business processes through which decision-makers are informed in context. With sim-IS training and wargaming environments, integrated business processes can be developed and evaluated iteratively, with the respective strengths of each approach and with the concurrent development of user competencies. While the cycle of research focuses on analysis in support of experimentation and capability and concept development, this envisioned cycle of training could support the synchronized development of ISs, organizational processes, and user competencies. RPA-based sim-IS environments present an opportunity for supporting such an approach to the coevolution of all three elements of joint cognitive systems.

This holistic approach to JCS coevolution is particularly pressing given the pursuit of AI-enabled decision-support systems. Chapter II identified the risks of out-of-the-loop-unfamiliarity and over-trust which can be expected to increase with the introduction of increasingly complex automated information systems. Simulation-supported environments are valuable tools for the development of autonomous and other AI-supported systems (Blais, 2016). They may also support the development and evaluation of processes and user competencies necessary to mitigate the trust and OOTLUF risks associated with the employment of AI-enabled ISs.

The introduction of new information technology is often intended to be conducted in tandem with the redesign of an organization's structure and/or processes. While organizations ideally redesign their structure and processes in conjunction with the implementation of new ISs, "managers often have only a general, amorphous knowledge of how they intend to use technology…and no one may be certain just what the job or the tasks will be" (Klein & Ralls, 1997, p. 342). In such situations, "job and organizational characteristics are treated not as predictors of training design, but as dependent variables influenced, in part, by technology training" (Klein & Ralls, 1997, p. 342). Just as wargaming environments facilitate analysts' exploration of a novel technology or concept in context, so can sim-IS environments support the exploration and coevolution of JCSs in context. In the envisioned cycle of training, RPA-based sim-IS environments present an opportunity for supporting enhanced staff training and the wargaming of new integrated business process designs.

APPENDIX A: SIM-IS DATA TRANSFORMATION COMPONENTS FOR TRANSLATION

Data Translation Consideration	RPA Data Transformation Component	Sim-IS RPA Module Which Uses this Component	Stage of Sim-IS Development for Addressing in Sim-IS Design	Special Considerations
Mapping Unit Instances	RPA Schedule	Extract Module & Entry Module	StartEx Data Generation (Extract/Entry Module development provides information for unique identifiers per system)	Unit Names may need to be adjusted in StartEx Data Generation to be unique (i.e., "Company A" may be insufficient as multiple units may possess a "Company A")
Mapping Entity Instances	Entity Status Tracker	Extract Module & Entry Module	StartEx Data Generation (Extract/Entry Module development provides information for unique identifiers per system)	This table is used differently for entity and aggregate simulations. The JLVC ID generated by the OBS may is useful, though in some instances (e.g., JCATS) the simulation generates its own unique identifier for entities
Mapping Supply Type Identifiers	RPA Schedule	Extract Module & Entry Module	StartEx Data Generation (Extract/Entry Module development provides information for unique supply type identifiers)	While IS often use NSNs, simulations sometimes use nomenclature. This may require adjusting simulation parametric data (e.g., JCATS' Vista, MTWS' APEX)
Mapping Supply Type Quantities	Supply Types Table	Translation Module	Sim-IS Requirements Specification & Translation	(Extract/Entry Module development provides requirements for

Data Translation Consideration	RPA Data Transformation Component	Sim-IS RPA Module Which Uses this Component	Stage of Sim-IS Development for Addressing in Sim-IS Design	Special Considerations
			Module Design	simulation and information systems' supply type quantities/attribute statuses)
Mapping Equipment/ Personnel Readiness Level	Entity Readiness Levels Table	Translation Module	Sim-IS Requirements Specification & Translation Module Design	(Extract/Entry Module development provides requirements for simulation and information systems' entity status levels)

APPENDIX B: IS INTERFACE DATA MODEL (CLC2S)

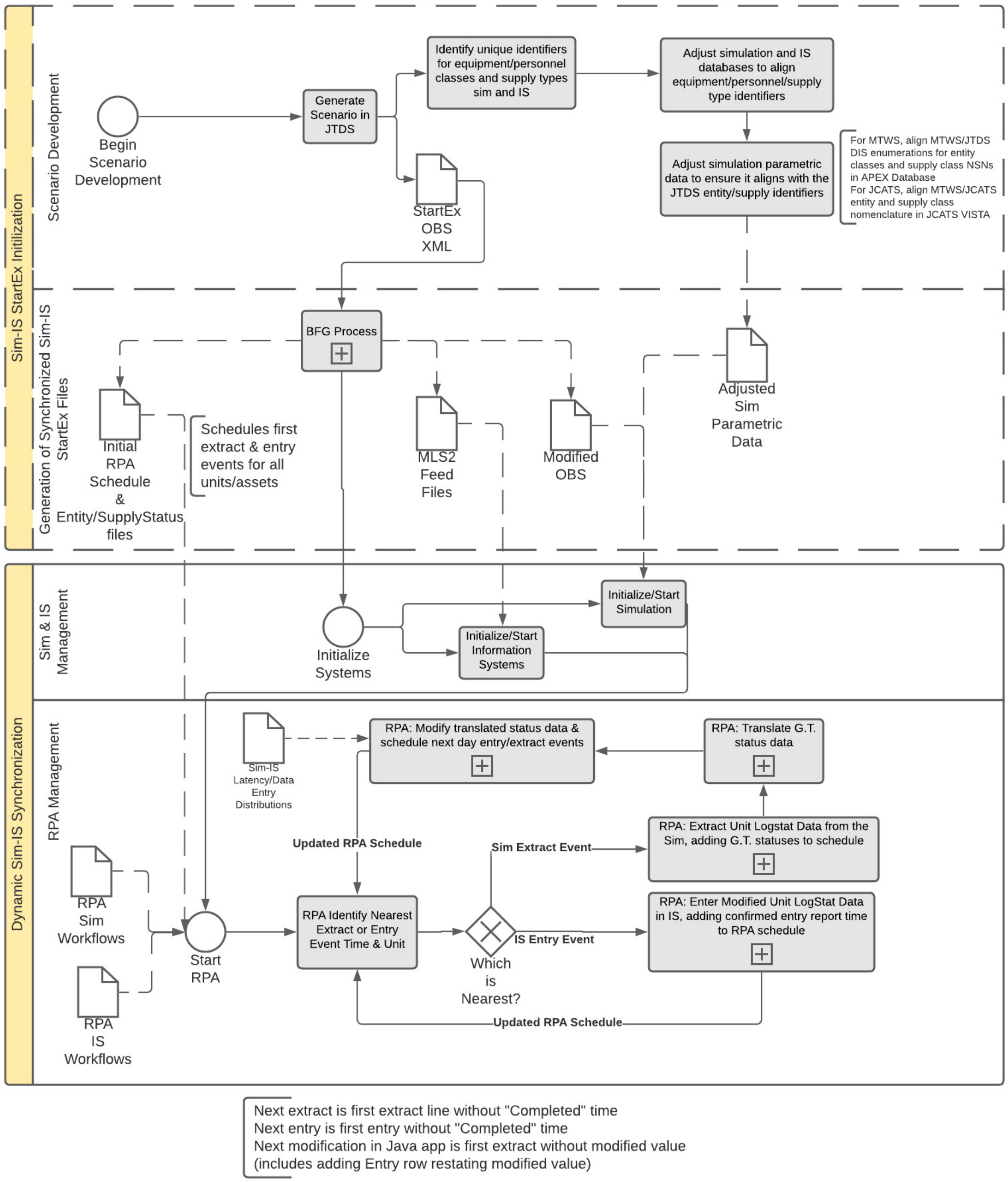
IS Interface Data Model (CLC2S)			
IS Element	Location in IS GUI	Data Type & Formatting Requirements	Data Transformation Component Source
Entry Event: Initial Unit Selection for Supply, Equipment, or Personnel Status Updates			
Key: Unit	User Profile -Change Unit -New Unit	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	For Supply Entry: Supply Status Tracker: “Unit_Name” For Entity Status Entry: Entity Status Tracker: “Unit_Name”
Entry Event: Supply Status Entry			
Key: Unit	Assets -Manage Supplies -Search Supplies -”Unit”	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	Supply Status Tracker: “Unit_Name”
Key: Supply Type	Assets -Manage Supplies -Search Supplies -”NSN”	Free text	Supply Status Tracker: “Supply_Type_NSN”
Value Entry Field	Assets -Manage Supplies -Modify Unit Supply Item -Inventory Information	Free text: Non-negative integer, less than 100,000,000	Supply Status Tracker: “Modified Status”
Entry Event: Equipment Readiness Status Update			
Key: Unit	Assets -Manage Equipment -Search Equipment -”Unit”	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	Entity Status Tracker: “Unit_Name”
Key: Equipment Serial #	Assets -Manage Equipment -Search Equipment -”Serial”	Free text*	Entity Status Tracker: “EDIPI_or_Serial_Number”
Value Entry Field	Assets -Manage Equipment -Modify Equipment	Drop-down menu (“Status”): Menu cannot be filtered. There are 7 options.	Entity Status Tracker: “Reported_Status”
Entry Event: Personnel Readiness Status Update			
Key: Unit	Assets -Manage Personnel -Search Personnel -”Unit”	Drop-down menu: Unit name must be selected from the options provided (mouse click). The menu can be filtered through free text entry of the unit name	Entity Status Tracker: “Unit_Name”
Key: First Name & Last Name	Assets -Manage Personnel -Search Personnel -”First Name” & “Last Name”	Free text: Two fields, “First Name” and “Last Name” **	Entity Status Tracker: “Personnel Names” Format: Last + ,” “ + First +” “+ Middle Initial

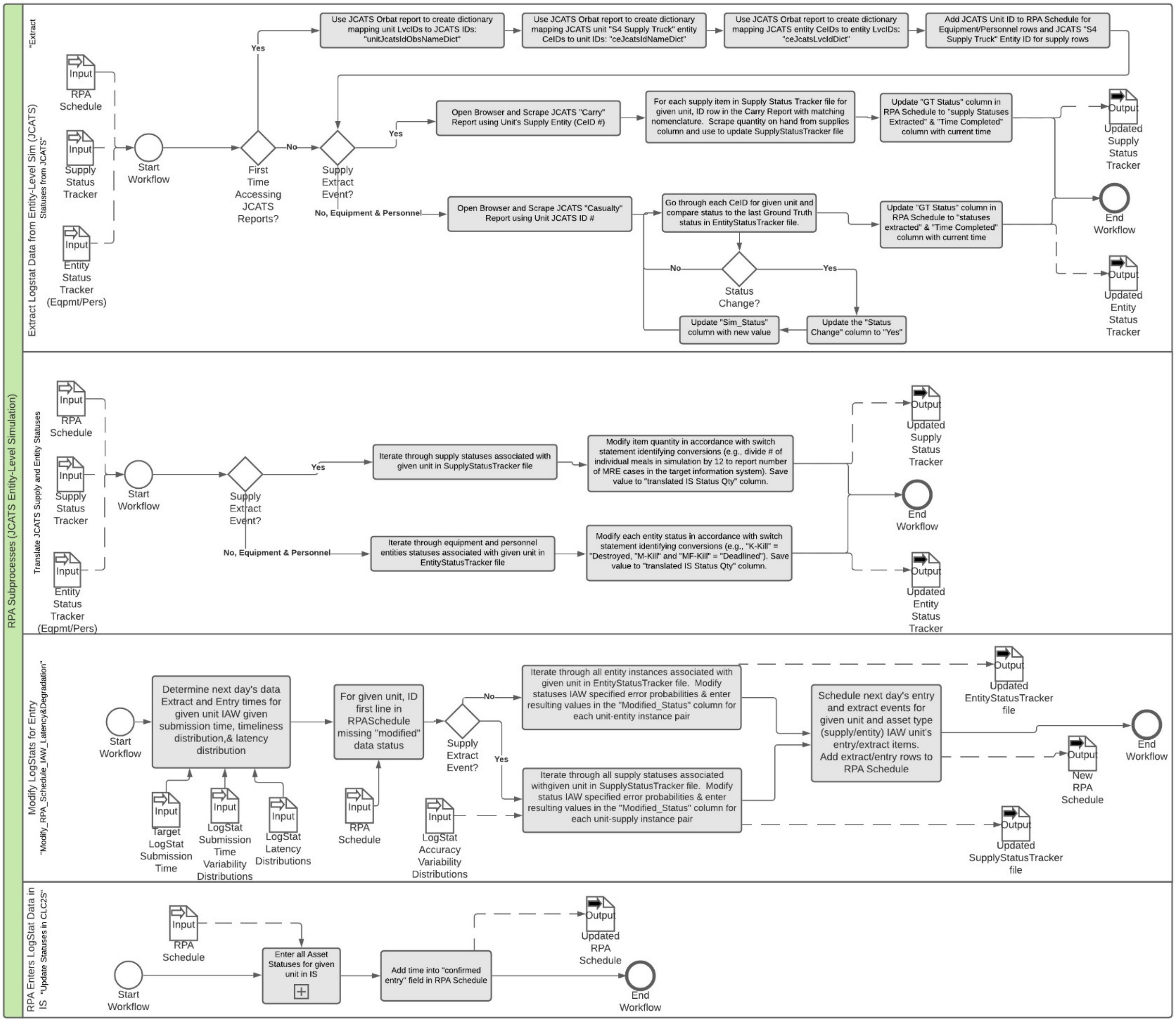
IS Interface Data Model (CLC2S)			
IS Element	Location in IS GUI	Data Type & Formatting Requirements	Data Transformation Component Source
Value Entry Field	Assets -Manage Personnel -Modify Personnel	Drop-down menu (“Capability”): Menu cannot be filtered. There are 24 options.	Entity Status Tracker: “Reported_Status”

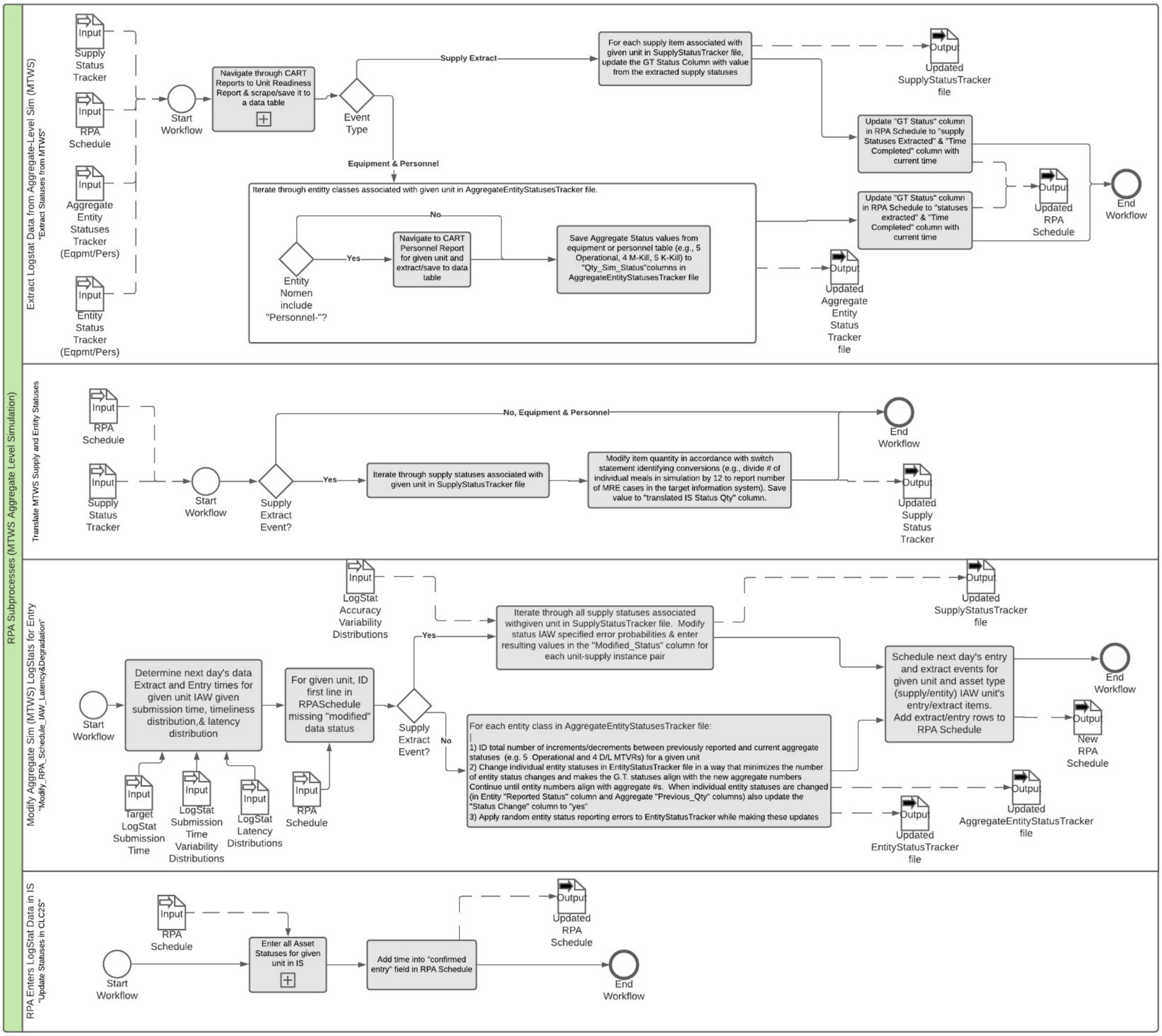
*If the sim-IS scenario is built so that unique serial numbers are assigned for all equipment entities, regardless of equipment type (BFG Application default), the equipment serial number is sufficient. Otherwise, an additional unique identifier for the equipment type (nomenclature, TAMCN, or NSN) will have to be selected for use along with the serial number to identifying unique equipment entity instances in CLC2S.

**If the sim-IS scenario is built so that no unit contains personnel with the same name (first and last), the First and Last Name fields are sufficient as a unique identifier for personnel. Otherwise, an additional identifier (e.g., ZAP, Sex, MOS) must be selected for use along with the personnel first and last names. However, unlike with equipment, none of these additional fields provides a guarantee that the correct individual will be selected, as EDIPI or Social Security Number are not options for personnel identification. It is therefore incumbent upon the sim-IS scenario developer to ensure personnel allocated to units are unique relative to the unique identifier attributes chosen.

APPENDIX C: RPA-BASED SIM-IS ARCHITECTURE CONCEPTUAL MODELS




RPA Subprocesses (JCATS Entity-Level Simulation)
Extract Logstat Data from Entity-Level Sim (JCATS) "Extract Statuses from JCATS"
Input
RPA Schedule
Input
Supply Status Tracker
Input
Entity Status Tracker (Eqpmt/Pers)
Start Workflow
First Time Accessing JCATS Reports?
Yes
No
Use JCATS Orbat report to create dictionary mapping unit LvcIDs to JCATS IDs: "unitJcatsIdObsNameDict"
Use JCATS Orbat report to create dictionary mapping JCATS unit "S4 Supply Truck" entity CeIDs to unit IDs: "ceJcatsIdNameDict"
Use JCATS Orbat report to create dictionary mapping JCATS entity CeIDs to entity LvcIDs: "ceJcatsLvcIdDict"
Add JCATS Unit ID to RPA Schedule for Equipment/Personnel rows and JCATS "S4 Supply Truck" Entity ID for supply rows
Supply Extract Event?
Yes
Open Browser and Scrape JCATS "Carry" Report using Unit's Supply Entity (CeID #)
For each supply item in Supply Status Tracker file for given unit, ID row in the Carry Report with matching nomenclature. Scrape quantity on hand from supplies column and use to update SupplyStatusTracker file
Update "GT Status" column in RPA Schedule to "supply Statuses Extracted" & "Time Completed" column with current time
Output
Updated Supply Status Tracker
No, Equipment & Personnel
Open Browser and Scrape JCATS "Casualty" Report using Unit JCATS ID #
Go through each CeID for given unit and compare status to the last Ground Truth status in EntityStatusTracker file.
Update "GT Status" column in RPA Schedule to "statuses extracted" & "Time Completed" column with current time
End Workflow
Status Change?
No
Yes
Update "Sim_Status" column with new value
Update the "Status Change" column to "Yes"
Output
Updated Entity Status Tracker
Translate JCATS Supply and Entity Statuses
Input
RPA Schedule
Input
Supply Status Tracker
Input
Entity Status Tracker (Eqpmt/Pers)
Start Workflow
Supply Extract Event?
Yes
Iterate through supply statuses associated with given unit in SupplyStatusTracker file
Modify item quantity in accordance with switch statement identifying conversions (e.g., divide # of individual meals in simulation by 12 to report number of MRE cases in the target information system). Save value to "translated IS Status Qty" column.
Output
Updated Supply Status Tracker
End Workflow
No, Equipment & Personnel
Iterate through equipment and personnel entities statuses associated with given unit in EntityStatusTracker file
Modify each entity status in accordance with switch statement identifying conversions (e.g., "K-Kill" = "Destroyed, "M-Kill" and "MF-Kill" = "Deadlined"). Save value to "translated IS Status Qty" column.
Output
Updated Entity Status Tracker
Modify LogStats for Entry "Modify_RPA_Schedule_IAW_Latency&Degradation"
Start Workflow
Determine next day's data Extract and Entry times for given unit IAW given submission time, timeliness distribution,& latency distribution
Input
Target LogStat Submission Time
Input
LogStat Submission Time Variability Distributions
Input
LogStat Latency Distributions
For given unit, ID first line in RPASchedule missing "modified" data status
Input
RPA Schedule
Supply Extract Event?
No
Yes
Iterate through all entity instances associated with given unit in EntityStatusTracker file. Modify statuses IAW specified error probabilities & enter resulting values in the "Modified_Status" column for each unit-entity instance pair
Iterate through all supply statuses associated withgiven unit in SupplyStatusTracker file. Modify status IAW specified error probabilities & enter resulting values in the "Modified_Status" column for each unit-supply instance pair
Input
LogStat Accuracy Variability Distributions
Output
Updated EntityStatusTracker file
Schedule next day's entry and extract events for given unit and asset type (supply/entity) IAW unit's entry/extract items. Add extract/entry rows to RPA Schedule
Output
New RPA Schedule
End Workflow
Output
Updated SupplyStatusTracker file
RPA Enters LogStat Data in IS "Update Statuses in CLC2S"
Input
RPA Schedule
Start Workflow
Enter all Asset Statuses for given unit in IS
Add time into "confirmed entry" field in RPA Schedule
Output
Updated RPA Schedule
End Workflow


RPA Subprocesses (MTWS Aggregate Level Simulation)
Extract Logstat Data from Aggregate-Level Sim (MTWS)
"Extract Statuses from MTWS"
Input
Supply Status Tracker
Input
RPA Schedule
Input
Aggregate Entity Statuses Tracker (Eqpmt/Pers)
Input
Entity Status Tracker (Eqpmt/Pers)
Start Workflow
Navigate through CART Reports to Unit Readiness Report & scrape/save it to a data table
Event Type
Supply Extract
Equipment & Personnel
For each supply item associated with given unit in SupplyStatusTracker file, update the GT Status Column with value from the extracted supply statuses
Output
Updated SupplyStatusTracker file
Update "GT Status" column in RPA Schedule to "supply Statuses Extracted" & "Time Completed" column with current time
Update "GT Status" column in RPA Schedule to "statuses extracted" & "Time Completed" column with current time
Output
Updated RPA Schedule
End Workflow
Iterate through entitty classes associated with given unit in AggregateEntityStatusesTracker file.
No
Yes
Entity Nomen include "Personnel-"?
Navigate to CART Personnel Report for given unit and extract/save to data table
Save Aggregate Status values from equipment or personnel table (e.g., 5 Operational, 4 M-Kill, 5 K-Kill) to "Qty_Sim_Status"columns in AggregateEntityStatusesTracker file
Output
Updated Aggregate Entity Status Tracker file
Translate MTWS Supply and Entity Statuses
Input
RPA Schedule
Input
Supply Status Tracker
Start Workflow
Supply Extract Event?
No, Equipment & Personnel
Yes
Iterate through supply statuses associated with given unit in SupplyStatusTracker file
Modify item quantity in accordance with switch statement identifying conversions (e.g., divide # of individual meals in simulation by 12 to report number of MRE cases in the target information system). Save value to "translated IS Status Qty" column.
End Workflow
Output
Updated Supply Status Tracker
Modify Aggregate Sim (MTWS) LogStats for Entry
"Modify_RPA_Schedule_IAW_Latency&Degradation"
Input
LogStat Accuracy Variability Distributions
Iterate through all supply statuses associated withgiven unit in SupplyStatusTracker file. Modify status IAW specified error probabilities & enter resulting values in the "Modified_Status" column for each unit-supply instance pair
Output
Updated SupplyStatusTracker file
Start Workflow
Determine next day's data Extract and Entry times for given unit IAW given submission time, timeliness distribution,& latency distribution
For given unit, ID first line in RPASchedule missing "modified" data status
Supply Extract Event?
Yes
No
Schedule next day's entry and extract events for given unit and asset type (supply/entity) IAW unit's entry/extract items. Add extract/entry rows to RPA Schedule
End Workflow
Output
New RPA Schedule
For each entity class in AggregateEntityStatusesTracker file:
1) ID total number of increments/decrements between previously reported and current aggregate statuses (e.g. 5 Operational and 4 D/L MTVRs) for a given unit
2) Change individual entity statuses in EntityStatusTracker file in a way that minimizes the number of entity status changes and makes the G.T. statuses align with the new aggregate numbers Continue until entity numbers align with aggregate #s. When individual entity statuses are changed (in Entity "Reported Status" column and Aggregate "Previous_Qty" columns) also update the "Status Change" column to "yes"
3) Apply random entity status reporting errors to EntityStatusTracker while making these updates
Output
Updated AggregateEntityStatusTracker file
Output
Updated EntityStatusTracker file
Input
Target LogStat Submission Time
Input
LogStat Submission Time Variability Distributions
Input
LogStat Latency Distributions
Input
RPA Schedule
RPA Enters LogStat Data in IS
"Update Statuses in CLC2S"
Input
RPA Schedule
Start Workflow
Enter all Asset Statuses for given unit in IS
Add time into "confirmed entry" field in RPA Schedule
Output
Updated RPA Schedule
End Workflow

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: EXAMPLE SIM-IS STARTEX DATA GENERATION PROCESS

A. ALIGNING JTDS SUPPLY CLASSES WITH JCATS PARAMETRIC DATA

1. Entities

For ensuring entities specified in JTDS align with the entity options within Vista, you need a list of all entity enumerations specified in JTDS. For each entity in JTDS, find the entity name in Vista.

If there is no Vista entity with the same exact nomenclature, regardless of NSN/ Enums, ID the closest platform and change the Vista entity name to exactly match the JTDS entity name:

For example, the JTDS M777 entity “HOW M777A2 TWD 155MM” (see Figure 67) had a closest Vista option of “M777A2 155MM TOWED HOWITZER.” The Vista entity name was changed to “HOW M777A2 TWD 155MM,” as seen in Figure 68.

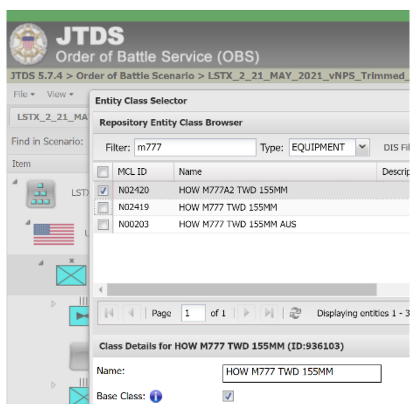


Figure 67. JTDS Entity Class Selection

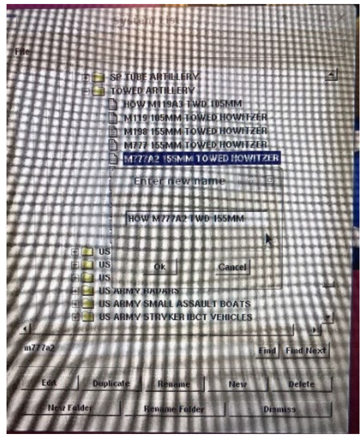


Figure 68. Changing an Entity Class Name in Vista

After aligning these names, the platform and associated munitions now manifest in the “System/Munition Report” discussed below.

2. Supplies

After all equipment and personnel entity classes to be used in a scenario are identified/synchronized across Vista and JTDS, the supply classes can be identified and synchronized.

The synchronization of munition classes to be used is more complex than fuel/water/rations, as it requires ensuring that all entity classes to be used in a scenario are using the same types of munition variants (supply classes). This can be approached one of two ways described below.

a. Munition Supply Class Alignment Method 1:

For each type of munition intended to be used in a scenario, (e.g., .50 Cal), a developer pulls up the “Munition Use” report for every potential .50Cal supply class (Figure 68). If entities to be used for a given scenario are associated with different .50Cal supply class variants, the entity class characteristics must be adjusted within Vista so that each entity uses the same .50Cal supply class.

For example, consider a scenario which includes both “HMMWV ARMORED M1116 M2” and “MRAP COUGAR JERRV 4X4 M2” entity classes. If it’s desired that a single .50Cal supply class be used to supply both of these entities, and the munition use reports are shown in Figure 69, then either of the two entity classes must be adjusted in Vista so they use a common .50Cal supply class in Vista.

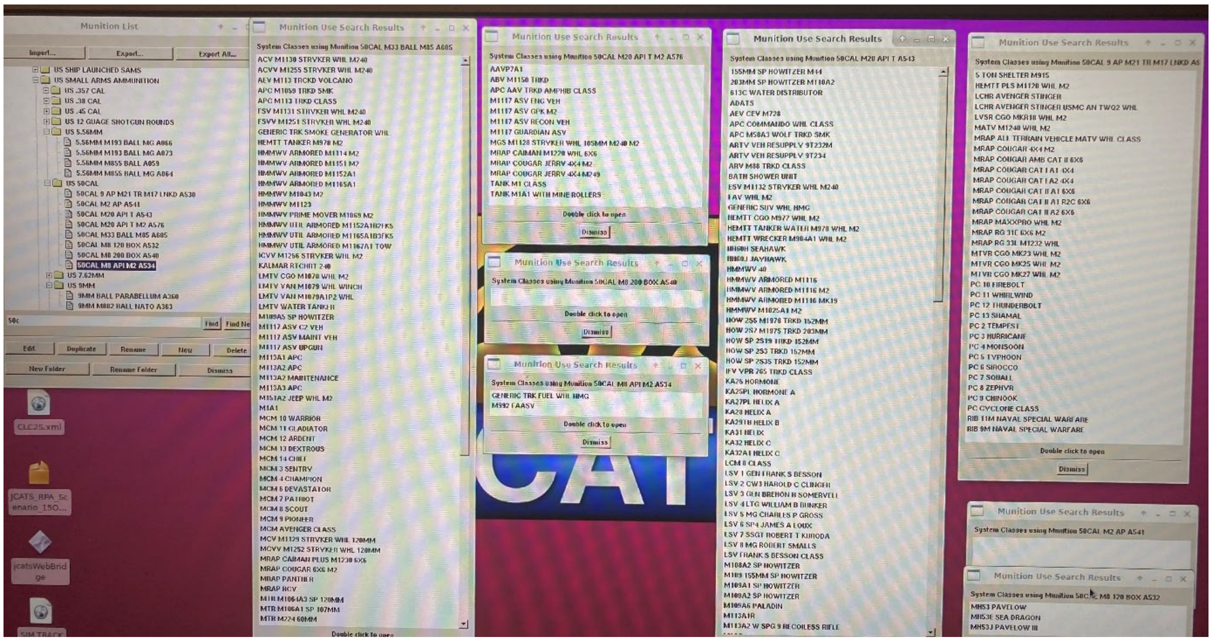


Figure 69. Vista “Munition Use” Reports for .50 Cal Supply Classes in Vista

If the “50CAL M20 API T A543” .50Cal munition variant is selected as the common munition for all entity classes using .50Cal rounds, and Vista entity classes have been adjusted accordingly, then that Supply Class can be added to the scenario in JTDS (Figure 70). The supply class nomenclature must match exactly between Vista and JTDS.

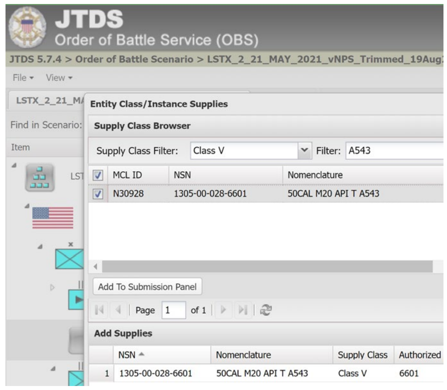
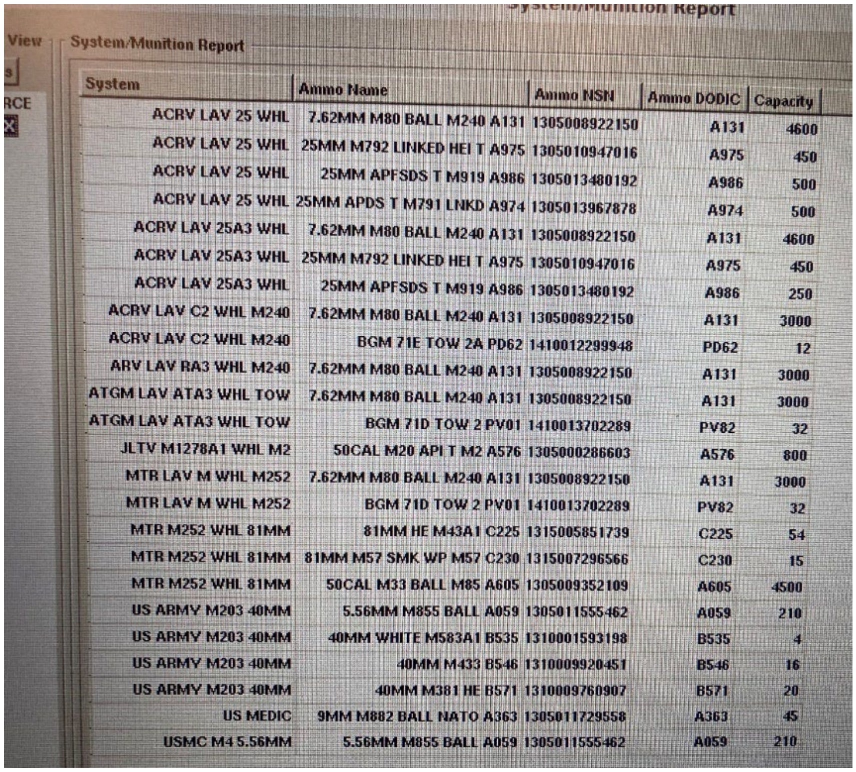


Figure 70. Selecting A543 .50Cal Supply Class in a JTDS Scenario

b. Munition Supply Class Alignment Method 2:

The other method for aligning munitions classes requires exporting an OBS XML from JTDS once all entity classes have been selected and added to a scenario in JTDS. After loading the XML into Vista with the associated fchar file, the developer can pull a System/Munition List report for the given Vista scenario (Figure 71). This report identifies all ammunition classes associated with a given scenario relative to their associated entity classes according to the entity class characteristics specified in the fchar file. This is a more thorough method for ensuring all entity classes for a scenario are using the same munition classes. The process for aligning ammunition class use across entity classes and then populating JCATS scenario accordingly follows the same process specified for method 1.


System/Munition Report

System	Ammo Name	Ammo NSN	Ammo DODIC	Capacity
ACRV LAV 25 WHL	7.62MM M80 BALL M240 A131	1305008922150	A131	4600
ACRV LAV 25 WHL	25MM M792 LINKED HEI T A975	1305010947016	A975	450
ACRV LAV 25 WHL	25MM APFSDS T M919 A986	1305013480192	A986	500
ACRV LAV 25 WHL	25MM APDS T M791 LNKD A974	1305013967878	A974	500
ACRV LAV 25A3 WHL	7.62MM M80 BALL M240 A131	1305008922150	A131	4600
ACRV LAV 25A3 WHL	25MM M792 LINKED HEI T A975	1305010947016	A975	450
ACRV LAV 25A3 WHL	25MM APFSDS T M919 A986	1305013480192	A986	250
ACRV LAV C2 WHL M240	7.62MM M80 BALL M240 A131	1305008922150	A131	3000
ACRV LAV C2 WHL M240	BGM 71E TOW 2A PD62	1410012299948	PD62	12
ARV LAV RA3 WHL M240	7.62MM M80 BALL M240 A131	1305008922150	A131	3000
ATGM LAV ATA3 WHL TOW	7.62MM M80 BALL M240 A131	1305008922150	A131	3000
ATGM LAV ATA3 WHL TOW	BGM 71D TOW 2 PV01	1410013702289	PV82	32
JLTV M1278A1 WHL M2	50CAL M20 API T M2 A576	1305000286603	A576	800
MTR LAV M WHL M252	7.62MM M80 BALL M240 A131	1305008922150	A131	3000
MTR LAV M WHL M252	BGM 71D TOW 2 PV01	1410013702289	PV82	32
MTR M252 WHL 81MM	81MM HE M43A1 C225	1315005851739	C225	54
MTR M252 WHL 81MM	81MM M57 SMK WP M57 C230	1315007296566	C230	15
MTR M252 WHL 81MM	50CAL M33 BALL M85 A605	1305009352109	A605	4500
US ARMY M203 40MM	5.56MM M855 BALL A059	1305011555462	A059	210
US ARMY M203 40MM	40MM WHITE M583A1 B535	1310001593198	B535	4
US ARMY M203 40MM	40MM M433 B546	1310009920451	B546	16
US ARMY M203 40MM	40MM M381 HE B571	1310009760907	B571	20
US MEDIC	9MM M882 BALL NATO A363	1305011729558	A363	45
USMC M4 5.56MM	5.56MM M855 BALL A059	1305011555462	A059	210

Figure 71. Using the "System/Munition Report" to Align Munition Supply Class Use Across Entity Classes

Entity-munition supply class alignments identified for the MCLOG OBS tested on 21 October include:

- 155MM: "HOW M777A2 TWD 155MM" entity class fires "155MM HOW M110 WP D550"
- 40MM: "US ARMY M203 40MM" entity class fires "40MM M381 HE B571"
- .50 Cal: "AAV AAVP7A1 TRKD AMPHIB" and "JLTV M1278A1 WHL M2" entity classes both fire "50CAL M20 API T M2 A576"

While it is clear that the nomenclature for munition classes must match between Vista and JCATS, NSN mismatch has not been tested to see if it would cause a problem when loading an XML in JCATS. Munition NSNs can be found in the Vista Munition Editor for munition class (Figure 72).

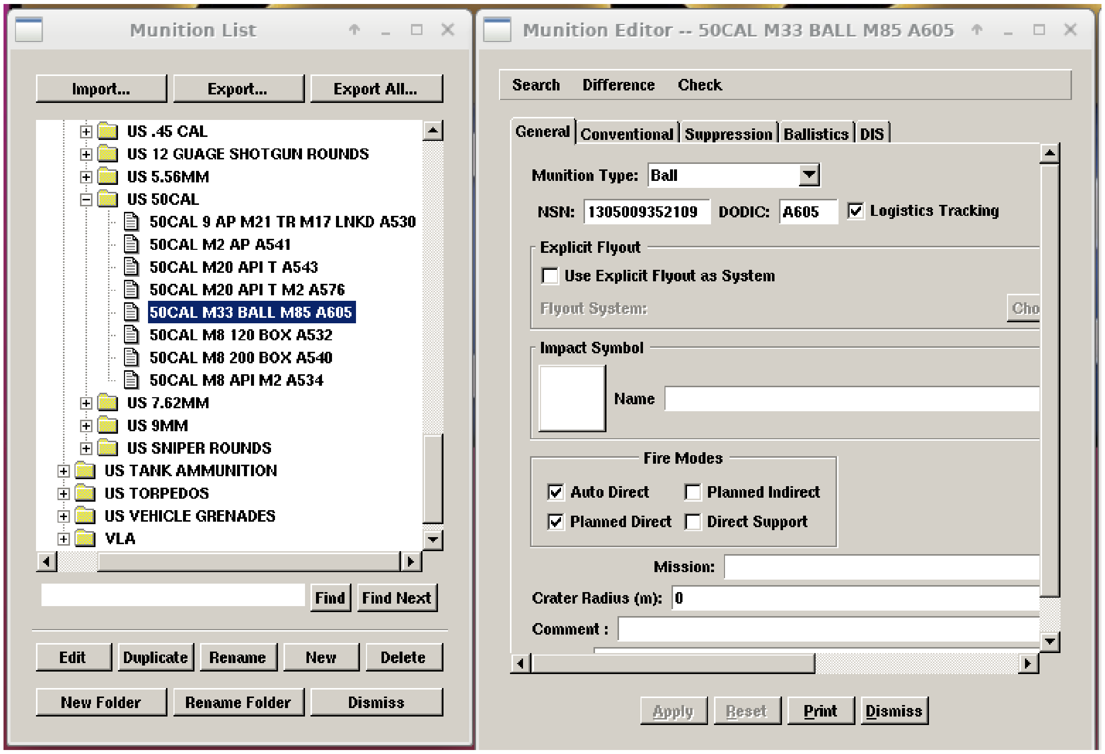


Figure 72. Vista Munition Editor

3. Aligning JTDS-Vista rations and water classes/NSNs

Rations and water within JCATS are always tracked as “Food” and “Water” respectively, though the NSNs for rations and water can be specified in Vista to align with the MREs NSN and water NSN being used for the scenario. These NSNs are specified within the “Supply Characteristics Editor” window within Vista, as illustrated in Figure 73. Adjusting these NSNs to match the NSNs for food (e.g., matching the Vista “Food” NSN to an individual “MRE” NSN to be used in JTDS) will ensure the respective food/ water supplies specified in JTDS will be decremented according to JCATS rations/water consumption.

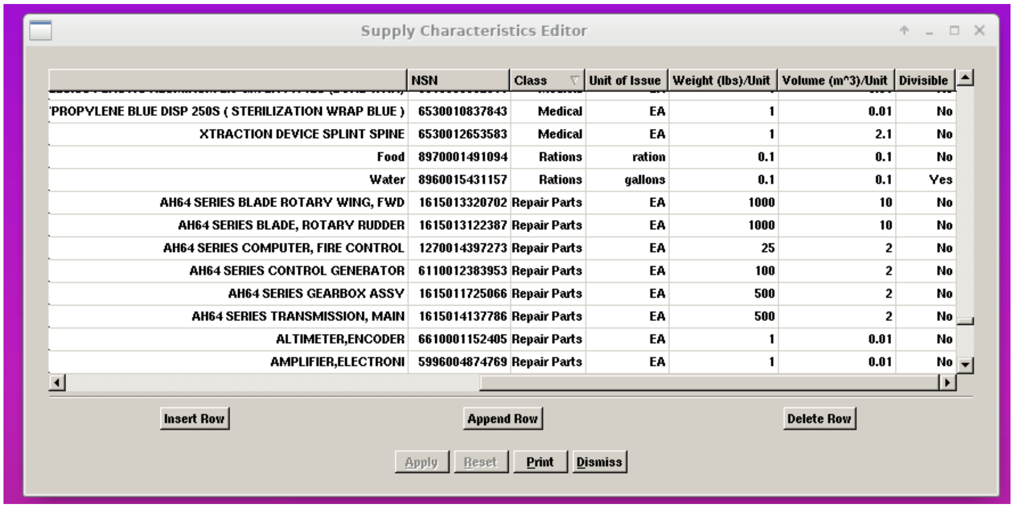


Figure 73. Vista Supply Characteristics Editor, Where NSNs May Be Adjusted

4. Aligning JTDS-Vista Fuel Class/NSNs

The NSN for fuel within JCATS is specified in the "Logistics Editor" window of Vista, as illustrated in Figure 74. Aligning this NSN with the NSN of the fuel supply class being used in JTDS will ensure that supply class is decremented in accordance with JCATS fuel consumption simulation.

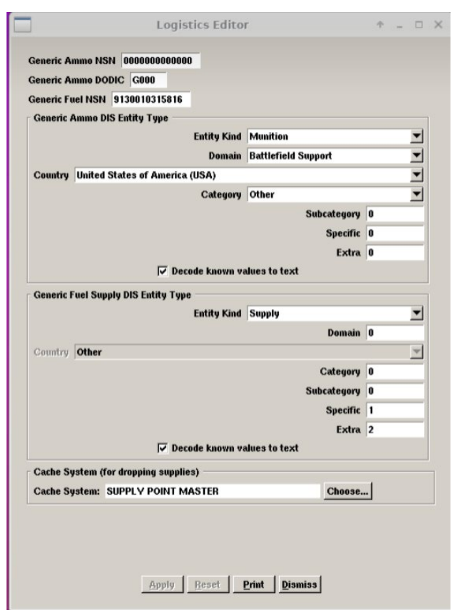


Figure 74. Vista Logistics Editor, Where Fuel NSN Is Modified

5. Assigning Supplies to Units under the S4 Supply Truck Entity

As an entity simulation, JCATS presents a challenge for reporting unit supplies. Unit logistics/supply personnel do not report the statuses of supplies that have been distributed to personnel (e.g., food, water, ammunition) and equipment (e.g., fuel). Instead, units report the supplies that are yet to be distributed and are held in consolidated locations. To simulate this dynamic, an entity in each unit can be established as the unit supply node to support reporting bulk supply stores.

To support this, an individual entity class must be selected, and the “carry capacity” specified for that entity class in Vista must be increased to a very high level.

For example, the entity “TRK TCTR HET M1070 WHL M240” may be selected for use as the unit supply node entity. In Vista -> Force Objects -> Systems, the entity’s carry capacity for weight, fuel, and volume can be adjusted to be very high. This ensures the S4 Supply Truck entity can carry all requisite supplies.

For a given scenario, the supplies for each unit must be allocated to a “TRK TCTR HET M1070 WHL M240” entity class type associated with each unit in JTDS. Each of these entities must also be given the same entity role name “S4 Supply Truck,” in order for the Batch File Generator (BFG) application to recognize that the supplies associated with the entity must be added to rows for the respective units in the supply feeds generated for CLC2S.

B. CONVERTING THE JTDS SCENARIO INTO SYNCHRONIZED JCATS, CLC2S, AND RPA STARTEX FILES

After building the JTDS scenario so it aligns with the Vista fchar data, run the OBS XML through the BFG.

Next, load the updated OBS XML into Vista to create a new scenario with the associated fchar file.

Before loading CLC2S feed files generated by the BFG, ensure “Root” is specified as higher for the senior-most unit.

1. Updating Supply Nomenclature for JCATS Reports

Before sim-IS robotic process automation (RPA) workflows can be executed, the SupplyStatusesFile must be updated to reflect the way supply type nomenclatures manifest in JCATS web reports. This can be done by manually pulling up JCATS Carry reports for units with all of the supply types being used for a scenario and then pasting those supply type nomenclature into the SupplyStatusFile adjacent to the respective supply type NSN. Figure 75 shows a JCATS web report (generated using the JCATS Web Bridge) on the right and the SupplyStatusesFile on the left. The NSNs for each supply item class can be found using Vista or JTDS.

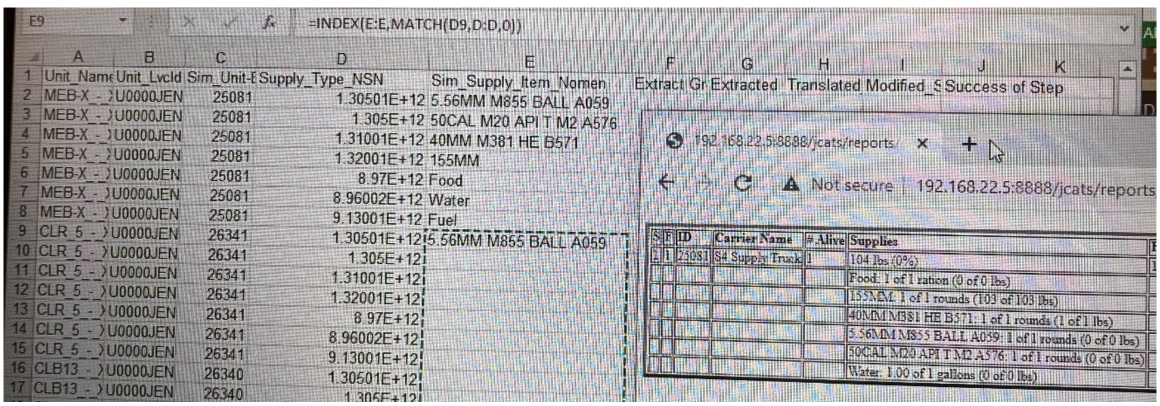


Figure 75. Updating the SupplyStatusesFile to Include Supply Item Nomenclatures Specified in the JCATS Carry Reports for Unit "S4 Supply Truck" Entities.

2. Updating Supply Class Unit of Issue Translation Ratios

The default setting for the translation of supply item units of issue from the simulation to the information system is one-to-one. If the translation needs to be something other than one-to-one, such as with MRE cases or for ammunition which is tracked within CLC2S by the box or case, the RPA switch function must be updated for the supply item type (see Figure 76).

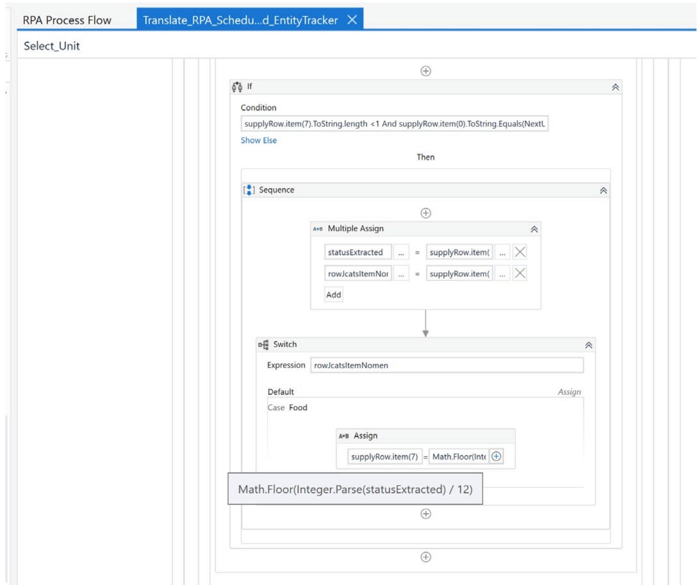


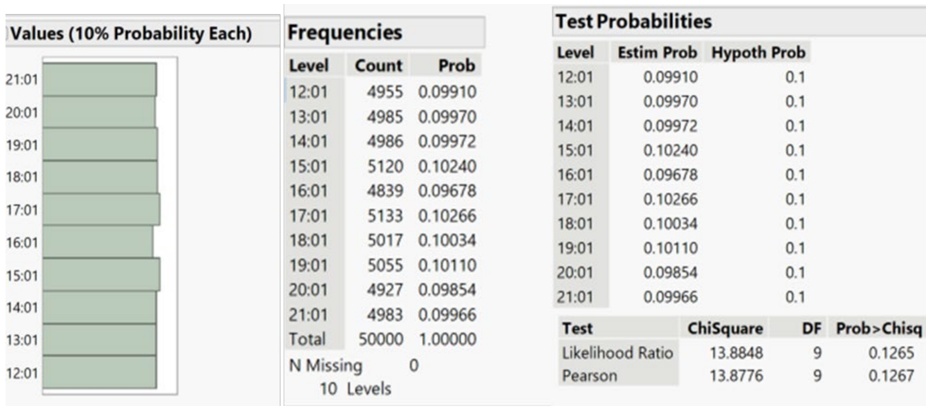
Figure 76. Adjusting the Conversion Ratio for Translating Each Supply Item Class's Unit of Issue From Simulation-To-Information System, Within the RPA Code.

APPENDIX E: TESTING THE UIPATH PSEUDO RANDOM NUMBER GENERATOR

Random number generators (RNG) provide an important capability for the simulation of different sim-IS information exchange dynamics. For temporal dynamics like information exchange latency and timeliness variability or for the entry of information into information systems, RNGs can be used to sample from target distributions to be simulated. Random numbers and pseudo random numbers can be generated using either RNGs or pseudo random number generators (PRNG), respectively. A primary difference between RNGs and PRNGs is the reproducibility of PRNG number series. Unlike RNGs, which generate random number series using some physical source, the values of any given PRNG-generated number series of pseudo random numbers are deterministic and are associated with a particular seed(Rukhin et al., 2010). Chi-square goodness of fit tests are a common tool for testing the randomness of both RNGs and PRNGs (Accardi & Gäbler, 2011; Ryabko et al., 2004; Shen, 2020).

> The [chi-square] distribution (i.e., a left skewed curve) is used to compare the goodness-of-fit of the observed frequencies of a sample measure to the corresponding expected frequencies of the hypothesized distribution.

The randomness of the PRNG organic to the UiPath RPA software was tested using the chi-square goodness of fit test. 50,000 samples were taken from ten values (times ranging from 12:01 to 21:01) with a uniform distribution. The goodness of fit of the resulting distribution of values was calculated relative to the uniform distribution using the chi-square goodness of fit test, with results shown in Figure 77. The observed p-value of 0.1267 is well below the minimum 0.95 level which would be necessary for supporting simulation of target temporal distributions with an alpha of 0.95.



Frequencies

Level	Count	Prob
12:01	4955	0.09910
13:01	4985	0.09970
14:01	4986	0.09972
15:01	5120	0.10240
16:01	4839	0.09678
17:01	5133	0.10266
18:01	5017	0.10034
19:01	5055	0.10110
20:01	4927	0.09854
21:01	4983	0.09966
Total	50000	1.00000

N Missing 0
10 Levels

Test Probabilities

Level	Estim Prob	Hypoth Prob
12:01	0.09910	0.1
13:01	0.09970	0.1
14:01	0.09972	0.1
15:01	0.10240	0.1
16:01	0.09678	0.1
17:01	0.10266	0.1
18:01	0.10034	0.1
19:01	0.10110	0.1
20:01	0.09854	0.1
21:01	0.09966	0.1

Test	ChiSquare	DF	Prob>Chisq
Likelihood Ratio	13.8848	9	0.1265
Pearson	13.8776	9	0.1267

Figure 77. Results of UiPath PRNG Randomness Testing

When testing random number generators, it is often necessary to also test to ensure the p-value is not too close to 1, which would suggest a perfect representation of the specified distribution (Accardi & Gäbler, 2011; Shen, 2020). This research is not concerned with representing the randomness in the sampling from target distributions, and so a one-sided testing of the chi-square goodness of fit was deemed sufficient for testing the UiPath PRNG. It is notable that while the observed latency and timeliness distributions were sufficiently random without being close to 1, the observed joint distributions which manifested in the RPA-generated RPA schedule files did align with the target joint distributions with a chi-square goodness of fit p-value of 1. As previously stated, this research is not concerned with simulating random noise that may be expected to prevent such perfect simulations of target distributions.

APPENDIX F: STRUCTURED INTERVIEW QUESTIONS

Before observing RPA in practice or Sim-IS initialization:

1) Please describe your organization's processes for the exchange of unit supply statuses and entity readiness statuses from constructive training simulations (e.g., JCATS, MTWS) and operational information systems (e.g., CLC2S, TCPT) in support of staff training environments.

    a. Please describe the manpower/time requirements for exchanging data between JCATS and CLC2S. Please describe the manpower/time requirements for exchanging data between MTWS and CLC2S.

    b. Please describe the possibility for unintentional human error in the manual exchange of data.

    c. Does your organization's manpower requirement for sim-IS information exchange limit the number of operational information systems which may be included in a staff training exercise (e.g., personnel management/casualty reporting system, SSDM, medical information system)?

2) How does your organization provide training for the Training and Readiness Tasks LOG-OPS-7002/8002 "Receive and Validate Support Requests," in the context of simulation-supported staff training environments?

    a. When imposing errors on the information presented to the training audience from the simulation, how does your organization determine the frequency with which to impose errors and when to do so?

    b. Does your organization have a system for imposing latency and timeliness variability on the information presented to the training audience, to prepare them for such dynamics in the real-world use of their information systems (e.g., CLC2S)? If so, please describe that system.

Following observation of the RPA-based Sim-IS information exchange prototype for JCATS-CLC2S and MTWS-CLC2S data exchange:

3) Does the RPA-based sim-IS information exchange prototype meet your organization's requirements for the exchange data between simulation and CLC2S for your organization's staff training environments? (timeliness, accuracy, precision, robustness)

4) Would the use of this sim-IS interoperability approach decrease manpower requirements for your organization's sim-IS environments for staff training exercises?

5) Would the use of this sim-IS interoperability approach allow your organization to introduce more information systems (e.g., personnel management, SSDM, medical information systems) in staff training exercises? Which additional information systems is your organization considering adding to staff training exercises?

6) Does the prototype simulate timeliness/latency distributions and/or error probabilities that would otherwise be too manpower intensive for your organization to achieve?

7) Does the demonstrated RPA-based sim-IS information exchange prototype's (3) reports meet your organization's requirements for informing exercise control personnel on the status of scheduled/completed sim-IS information exchanges?

8) Based on your organization's processes for information exchange between MTWS and CLC2S in support of exercises, how does the utility of the RPA-based approach compare to the utility of the recently fielded MTWS-CLC2S information exchange function?

9) Based on your organization's processes for information exchange between MTWS and CLC2S in support of exercises, how does the usability of the RPA-based approach compare to the usability of the recently fielded MTWS-CLC2S information exchange function?

10) Based on your organization's processes for information exchange between JCATS and CLC2S in support of exercises, how does the utility of the RPA-based approach compare to the utility of the recently fielded JCATS-CLC2S information exchange function?

11) Based on your organization's processes for information exchange between JCATS and CLC2S in support of exercises, how does the usability of the RPA-based approach compare to the usability of the recently fielded JCATS-CLC2S information exchange function?

Following observation of the initialization for Sim-IS information exchange prototype for JCATS-CLC2S and MTWS-CLC2S data exchange:

12) In regard to JCATS-CLC2S scenario initialization, is the initialization of the sim-IS environment for the RPA prototype more/less complex or manpower intensive for your organization than conventional sim-IS initialization processes?

13) In regard to MTWS-CLC2S scenario initialization, is the initialization of the sim-IS environment for the RPA prototype more/less complex or manpower intensive for your organization than conventional sim-IS initialization processes?

14) What obstacles exist which would need to be overcome for the design of sim-IS environments which include timeliness/latency distributions and errors automatically occurring with specified probabilities? What opportunities?

Please identify, from an organizational perspective, any potential additional uses of this RPA-based sim-IS information exchange methodology and middleware, and instances in which it may impact the design and development of simulation-supported staff training exercises?

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G: FEDERAL RPA COMMUNITY OF PRACTICE CONCEPTUAL MODELING SURVEY

RPA Conceptual Modeling Survey

Federal RPA Community of Practice Member Organizations

Survey Purpose:

This survey is intended to develop an understanding of current practices for conceptual modeling in support of RPA workflow design and maintenance. Feedback from this survey will help inform the development of RPA conceptual modeling best practices for the Federal RPA CoP and support PhD dissertation research relating to RPA.

While survey responses are anonymous, responses will be made available to Federal RPA CoP members. Survey participants have the option of making the name of their organization available to Federal RPA CoP members in association with their survey responses. This will allow collaboration among Federal RPA CoP members regarding different conceptual modeling languages and practices. Survey responses will be consolidated and anonymized for all survey participants who opt to keep their organization's name anonymous.

Defining Conceptual Models:

Here, conceptual model refers to platform-independent documentation of planned/fielded information systems and associated processes. Conceptual models are used to support design and maintenance of the processes, information systems, and/or user training.

Conceptual models can be developed for any audience, including engineers, managers, or system users.

Conceptual models can be developed using a variety of formats. Highly structured languages and diagrams (e.g., Unified Modeling Language (UML), SysML) are often used to ensure sufficient accuracy among engineers. Narrative and scenario-based models have broader accessibility for non-engineers (e.g., Multi-Viewpoint Conceptual Modeling [MVCM]). Graphical diagrams and tables may be used across both of these to help convey sociotechnical processes and data exchange dynamics.

In a field related to RPA, some developers of information systems and associated integrated business processes have found that extending business process modeling notation (BPMN) with user interface description languages helps capture all of the requisite dynamics for their integrated business processes.

Examples of Conceptual Modeling Formats and Languages Include:

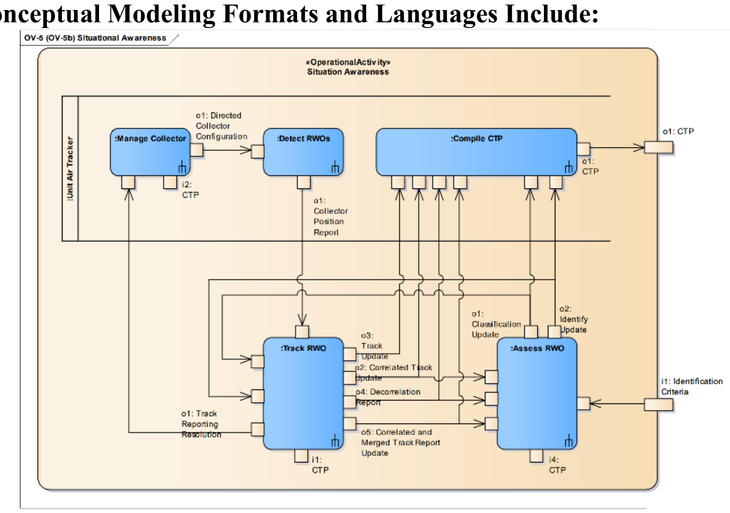


Figure 1. Example of a structured modeling notation (UML). From "Simulation environment architecture development using the DoDAF," by van den Berg, T.V. and Lutz, R., 2015, Fall SISO Simulation Interoperability Workshop, p. 7.

Narrative	Source	Behavior	Sink	Notes / Questions
Then, at 2:08 a.m. Thursday, Bosnia time, an F-16 from O'Grady's squadron patrolling in the area of the shootdown heard Basher 52 come up on his radio.	IP Radio	Communicate	FAC-A Radio	
It took about 12 minutes for procedures that positively identified O'Grady, and from there things moved rapidly.	FAC-A Radio	Communicate	IP Radio	
	IP Radio	Communicate	FAC-A Radio	
	FAC-A	Identify	IP	
	FAC-A Radio	Communicate	IP Radio	Requesting location and condition
	IP Radio	Communicate	FAC-A Radio	Reporting location and condition
	FAC-A Radio	Communicate	RescueTeam Radio	Reporting location and condition
	FAC-A Radio	Communicate	RESCORT Radio	Reporting location and condition
	FAC-A Radio	Communicate	Recovery Vehicle Radio	Reporting location and condition
At 5 a.m., six minutes before dawn, the rescue team began lifting off the Kearsarge.	Recovery Vehicle	Maneuver		
	RESCORT	Maneuver		
At 5:50, the rescue aircraft made their run: two CH-53 Sea Stallion helicopters with two dozen hand-picked Marines under 1st Lt. Martin Wetterauer; escort Cobra AH1W helicopters under Maj. Scott Mikyleby, and above them a pair of Harrier jump jets led by Maj. Michael Ogden.	Recovery Vehicle	Maneuver		
	RESCORT	Maneuver		
	RESCORT Radio	Communicate	RescueTeam Radio	
	RESCORT Radio	Communicate	Recovery Vehicle Radio	

Figure 2. Example of "Narrative View" of a MVCM model. From Multi-Viewpoint Conceptual Modeling [Conference presentation], by Morse, K.L. and Drake, D.L., 2020, SISO CA2X2 Forum, p. 11.

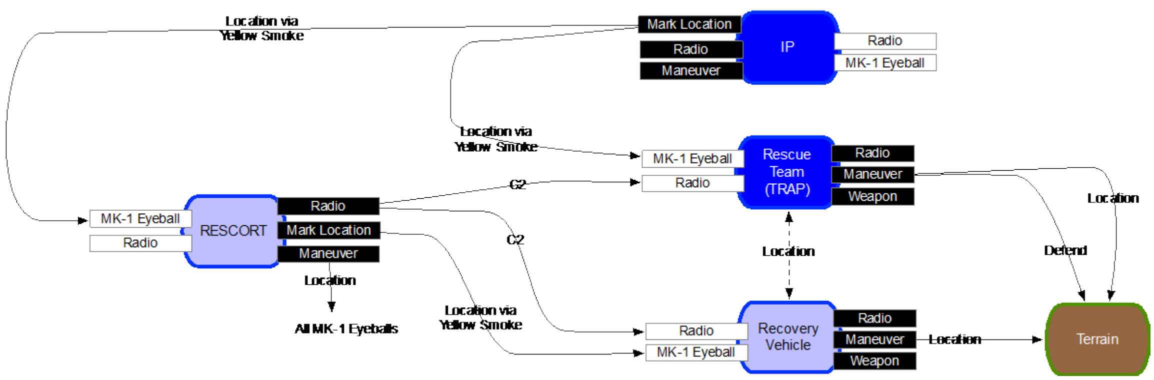


Figure 3. Example of Entity Subset Conceptual Model of a MVCM model. From Morse K.L. and Drake D.L. (2020), p. 15.

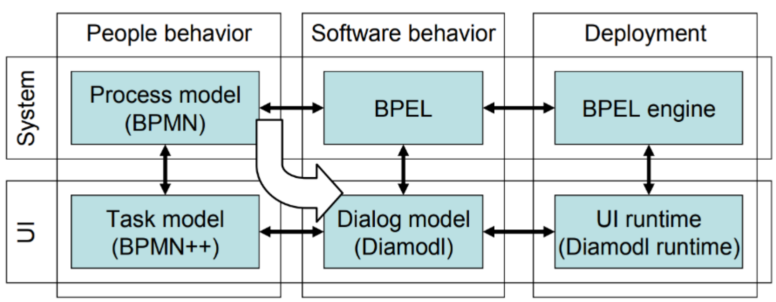


Figure 4. Example business process and UI modeling notations. From "Enhancing the Usability of BPM-Solutions by Combining Process and User-Interface Modelling," by Trætteberg, H. and Krogstie, J., 2008, p. 93.

This survey asks about two levels of conceptual models:
RPA Conceptual Model: Provides documentation of an RPA workflow itself, supporting workflow design, development, and maintenance as associated processes and information systems/formats evolve. These conceptual models document details about interactions with user interfaces, RPA data manipulation, and requisite data formats for information systems with which the RPA interacts. These conceptual models are RPA platform-independent, so while they identify requisite GUI attributes/anchors/data formats for RPA interactions, they are not specific to a particular RPA platform (e.g., UiPath, Blue Prism) and therefore support transition of the RPA workflows to other platforms if necessary.

Integrated Business Process (or Business Process-Information System [BP-IS]) Conceptual Model: Provides documentation of integrated business processes to be supported by an RPA workflow. This conceptual model supports the design and continuous development (coevolution) of the business process and associated information systems (including RPA workflows). It also informs training organizations regarding requisite human competencies for operating within the integrated business process.

1. What is your organization's name? ________________________________

2. Can your survey responses be made available to all members of the Federal RPA Community of Practice? Yes / No

3. Can your organization's name be associated with the survey responses provided? If yes, the responses will only be associated with the organization, not with any individuals associated with the survey. If no, the survey responses will be completely anonymous relative to their originating organization. Yes / No

4. How many years has RPA been used within your organization? ________

5. How does your organization leverage RPA?
   a. Currently, we are only researching potential uses
   b. Back-office process automation
   c. Developing reports
   d. Automating correspondence (e.g., emails)
   e. Supporting staff training environments or exercising integrated business processes
   f. Other: ____

6.1. How does your organization document integrated business processes to be supported with RPA?
   a. Prior documentation isn't necessary. RPA developers work side by side with users to design workflows directly in the RPA development platform on site.
   b. Flow diagrams are received from the users in assorted notations (BPMN, BPEL, flow diagrams etc.), and our RPA developers use them to build RPA workflows.
   c. Our standardized BP-IS conceptual models provide a high-level documentation of actions and UI interactions.
   d. Our standardized BP-IS conceptual models provide detailed documentation of user interface interactions and data formatting requirements.
   e. Our BP-IS conceptual models achieve both B and C, using a notation called: ______________
   f. Other:____

6.2: Notations/languages used include:___________

7.1. How does your organization develop and maintain RPA conceptual models?
   a. The only documentation is the RPA workflow documentation provided by the RPA development platform: _____________________
   b. RPA conceptual models provide a high-level documentation of actions and UI interactions, using a notation called: ______________
   c. RPA conceptual models provide details of user interface interactions and data formatting requirements, using a notation called: ______________
   d. RPA conceptual models achieve both B and C, using a notation called: ______________
   e. Other: ____

7.2: Notations/languages used include:___________

8.1. Does your organization have a defined process for updating both the RPA conceptual model and conceptual model for the associated integrated business process as they change over time?
   a. No. We don't maintain RPA conceptual models associated with BP-IS conceptual models
   b. No. We maintain models of RPA workflows and associated business processes, but we only maintain them through development of the RPA workflows.

c. Yes. We maintain and update models of RPA workflows and associated business processes over time.

8.2. Please provide a brief description of your organization's process.

9. If you answered yes to Question 5, is your organization satisfied that its conceptual modeling methodology for RPA and BP-IS is adequate for both design and continuous improvement and coevolution of both integrated business processes and associated RPA workflows?

a. Yes

b. No. Please explain:____

10. Does your organization use an RPA conceptual modeling notation that is platform independent?

a. Yes

b. No

c. Don't know

11. If you answered yes to question 8, please explain how your RPA conceptual modeling notation addresses the issue of different RPA platforms supporting different functions?

12. Does your organization use a standard technical RPA workflow language for the automated translation of RPA workflows between RPA platforms? If so, please describe/identify it. _____________

References:

van den Berg, T.V., & Lutz, R. (2015). Simulation environment architecture development using the DoDAF, Fall SISO Simulation Interoperability Workshop.

Morse, K.L. & Drake, D.L. (2020, September 22). *Multi-Viewpoint Conceptual Modeling: In Support of Simulation Interoperability Readiness Levels (SIRLs)*. [Conference Presentation]. NATO CA2X2 Forum 2020, Virtual Event.

Trætteberg, H., & Krogstie, J. (2008). Enhancing the Usability of BPM-Solutions by Combining Process and User-Interface Modelling. *PoEM*, p. 86-97.

APPENDIX H: DSEEP OVERLAY RECOMMENDATIONS

Step 1: Define Simulation Environment Objectives

Activity 1.1: Identify User/Sponsor Needs

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Identify requisite sim-IS environment participant perspectives for the intended use (Issue 1.1.1)
- Identify initial list of operational ISs to be represented in sim-IS environment and how they relate to participant perspectives (Issue 1.1.2)
- Identify the STS dynamics to be represented, for what purpose(s), and requisite levels of fidelity (Issue 1.1.3)

Sim-IS Environment-Specific Outcomes:

- Description of intended use of the sim-IS environment and sim-IS environment participant perspectives to be supported (Issue 1.1.1)
- Initial list of operational ISs to be represented in sim-IS environment, including function and form requirements for each (Issue 1.1.2)
- Initial list of STS information exchange dynamics to be represented and requisite levels of fidelity (Issue 1.1.3)

Activity 1.2: Develop Objectives

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Refine requirements for operational ISs to be represented (Issue 1.2.1)

- Evaluate generalizability of sources informing the design of integrated business processes and associated STS dynamics to be simulated (Issue 1.2.2)

Sim-IS Environment-Specific Outcomes:

- List of appropriate sources for informing the design of integrated business processes and STS dynamics and description of the degrees to which processes are standardized across the organization (Issue 1.2.2)

Activity 1.3: Conduct Initial Planning

No sim-IS environment-specific input, task, or outcome identified

Step 2: Perform Conceptual Analysis

Activity 2.1: Develop Scenario

Sim-IS Environment-Specific Inputs:

- CMs for real-world integrated business processes to be simulated (Issue 2.1.2)

- Sources for STS dynamics to be simulated (Issue 2.1.3)

Sim-IS Environment-Specific Tasks:

- Identify information to be presented for each participant perspective (Issue 2.1.1)

- Acquire/develop CMs for integrated business processes to be simulated (Issue 2.1.2) and documentation of associated STS dynamics (Issue 2.1.3)

Sim-IS Environment-Specific Outcomes:

- List of information to be presented for each perspective (Issue 2.1.1)

- CMs for target integrated business processes and STS dynamics (Issues 2.1.2, 2.1.3), including associated information presentation mediums

Activity 2.2: Develop Simulation Environment Conceptual Model

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Capture target integrated business processes in sim-IS environment conceptual model (Issue 2.2.1)
- Define how STS dynamics are to be simulated in the sim-IS environment (Issue 2.2.2)
- Model procedures for information entry into operational ISs through GUIs and other interfaces as appropriate (Issue 2.2.3)

Sim-IS Environment-Specific Outcomes:

- Sim-IS environment conceptual model that includes representation of processes through which ground truth information is presented to different sim-IS environment participant perspectives and associated STS dynamics (Issues 2.2.1, 2.2.2)
- Detailed models for the procedures through which information is delivered to/entered into operational ISs, aligned with the sim-IS environment conceptual model (Issue 2.2.3)

Activity 2.3: Develop Simulation Environment Requirements

Sim-IS Environment-Specific Inputs:

- List of operational ISs to be represented in the sim-IS environment and their requisite form and function

Sim-IS Environment-Specific Tasks:

- Define requirements for presentation of information to ISs (Issue 2.3.1)
- Define requirements for simulation of STS dynamics and performance of sim-IS information exchange mechanisms (Issue 2.3.2, 2.3.3))

- Define collection plans for documenting sim-IS environment participant procedures and observed STS dynamics (Issues 2.3.4, 2.3.5)

- Define requirements for management of RPA software (Issue 2.3.6)

Sim-IS Environment-Specific Outcomes: No additions identified

Step 3: Design Simulation Environment

Activity 3.1: Select Member Applications

Sim-IS Environment-Specific Inputs:

- Conceptual models for potential member applications (Issue 3.1.2)

- List of available code and parametric data repositories for potential member applications, including RPA modules and operational IS databases

Sim-IS Environment-Specific Tasks:

- Include ISs and IS information exchange mechanisms as member applications (Issue 3.1.1)

- Acquire/develop conceptual models for candidate member applications (Issue 3.1.2)

- Determine availability of RPA software platforms and RPA employment options (Issue 3.1.3)

Sim-IS Environment-Specific Outcomes:

- Conceptual models for all selected member applications (Issue 3.1.2)

Activity 3.2: Design Simulation Environment

Sim-IS Environment-Specific Inputs:

- Selected member applications' conceptual models (Issues 3.2.1, 3.2.2)

- Available sim-IS information exchange architecture designs and associated RPA modules, where appropriate (Issue 3.2.3)

Sim-IS Environment-Specific Tasks:

- Divide simulation of integrated business processes across sim-IS environment member applications (Issue 3.2.1)
- Specify information exchange mechanisms to be leveraged for different information exchanged between simulations and ISs (Issue 3.2.2)
- Select/design sim-IS information exchange architecture(s) (Issue 3.2.3)

Sim-IS Environment-Specific Outcomes:

- Initial sim-IS information exchange architecture design(s) (Issue 3.2.3)

Activity 3.3: Design Member Applications

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Identify how information presented in simulation interfaces maps to requisite ISs for presentation (Issue 3.3.1)
- Identify how information presented in simulations must be modified to simulate specified STS dynamics (Issue 3.3.2)
- Design sim-IS information exchange mechanism(s) in accordance with the sim-IS environment conceptual model and Issues 3.3.1 and 3.3.2, including in support of schedule and/or event-based sim-IS information exchanges (Issue 3.3.3)

Sim-IS Environment-Specific Outcomes: No additions identified

Activity 3.4: Prepare Detailed Plan

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Define plan for sim-IS integration (Issue 3.4.1)
- Define collection plans for observed organizational processes and STS dynamics (Issues 3.4.2, 3.4.3)

Sim-IS Environment-Specific Outcomes: No additions identified

Step 4: Develop Simulation Environment

Activity 4.1: Develop Simulation Data Exchange Model

Sim-IS Environment-Specific Inputs:

- Existing data transformation component files for RPA-based sim-IS information exchange (if RPA-based sim-IS information exchange approach is selected as a member application)

Sim-IS Environment-Specific Tasks:

- Design sim-IS data exchange requirements (Issue 4.1.1)

Sim-IS Environment-Specific Outcomes:

- Updated data transformation component files for RPA-base sim-IS information exchange architecture, as appropriate

Activity 4.2: Establish Simulation Environment Agreements

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Identify simulation and ID database elements that must be consistent or mapped (Issue 4.2.1)
- Design RPA modules for information extraction from simulation(s) (Issue 4.2.2)

- Identify authoritative data source(s) for ISs and STS dynamics (Issue 4.2.3)
- Define process for automating generation and synchronization of sim-IS scenario initialization files (Issue 4.2.4)
- Ensure the scenario save/restore plan includes sim-IS information exchange considerations (Issue 4.2.5)
- Identify requisite number of RPA bot instances for sim-IS information exchange (Issue 4.2.6)

Sim-IS Environment-Specific Outcomes:

- Initial design of RPA modules for information extraction from simulation(s), aligned with sim-IS environment conceptual model and selected RPA-based sim-IS information exchange architecture
- Initial design of process for automating generation and synchronization of sim-IS scenario initialization files

Activity 4.3: Implement Member Application Designs

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Implement development of IS databases and RPA data transformation component files (Issue 4.3.1)

Sim-IS Environment-Specific Outcomes:

- New or modified RPA modules for sim-IS information exchange and IS databases and RPA data transformation component files, as required

Activity 4.4: Implement Simulation Environment Infrastructure

No sim-IS environment-specific input, task, or outcome identified

Step 5: Integrate and Test Simulation Environment

Activity 5.1: Plan Execution

No sim-IS environment-specific input, task, or outcome identified

Activity 5.2: Integrate Simulation Environment

No sim-IS environment-specific input, task, or outcome identified

Activity 5.3: Test Simulation Environment

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Verify that the number of RPA bot instances is adequate (Issue 5.3.1)
- Test sim-IS environment StartEx data generation and synchronization process (Issue 5.3.2)
- Rehearse management of RPA modules and access to data transformation component files during sim-IS environment execution (Issue 5.3.3)

Sim-IS Environment-Specific Outcomes:

- Updates for RPA module performance data documentation, including speed, accuracy, and precision of modules

Step 6: Execute Simulation

Activity 6.1: Execute Simulation

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Conduct sim-IS environment data collection for observed organizational integrated business processes and associated STS dynamics (Issue 6.1.1)

Sim-IS Environment-Specific Outcomes:

- Each sim-IS environment executed with real-world units provides an additional reference for refining integrated business process conceptual models and STS dynamics, and additional insight into the degree to which defined processes are standardized across the enterprise.

Activity 6.2: Prepare Simulation Environment Outputs

No sim-IS environment-specific input, task, or outcome identified

Step 7: Analyze Data and Evaluate Results

Activity 7.1: Analyze Data

Sim-IS Environment-Specific Inputs: No additions identified

Sim-IS Environment-Specific Tasks:

- Analyze sim-IS environment representation of JCS(s) (Issue 7.1.1)

Sim-IS Environment-Specific Outcomes:

- Identification of notable divergences between observed integrated business processes/STS dynamics and those simulated by the sim-IS environment

Activity 7.2: Evaluate and Feedback Results

Sim-IS Environment-Specific Inputs:

- Analysis of observed integrated business processes and STS dynamics compared to existing conceptual models and STS dynamic estimates used to inform design of the sim-IS environment

Sim-IS Environment-Specific Tasks:

- Determine generalizability of observed divergence of integrated business processes and associated STS dynamics and ramifications for sim-IS environment redesign (Issue 7.2.1)

- Update conceptual model(s) for target integrated business processes, STS dynamics, and the sim-IS environment conceptual model and requisite member applications based on lessons learned from observing JCSs in sim-IS environment context (Issue 7.2.1)

Sim-IS Environment-Specific Outcomes:

- Updated conceptual models for integrated business processes, sim-IS environment, and member applications

- Updated STS dynamic estimates to be represented in future sim-IS environments

LIST OF REFERENCES

Accardi, L., & Gäbler, M. (2011). Statistical analysis of random number generators. In L. Accardi, W. Freudenberg, & M. Ohya, *Quantum Bio-Informatics IV* (Vol. 28, pp. 117–128). World Scientific. https://doi.org/10.1142/9789814343763_0009

Alaeddini, M., Asgari, H., Gharabi, A., & Rad, M. R. (2017). Leveraging business-IT alignment through enterprise architecture—an empirical study to estimate the extents. *Information Technology and Management*, *18*(1), 55–82.

Alcivar, I., & Abad, A. G. (2016). Design and evaluation of a gamified system for ERP training. *Computers in Human Behavior*, *58*, 109–118.

Al-Mashari, M., Al-Mudimigh, A., & Zairi, M. (2003). Enterprise resource planning: A taxonomy of critical factors. *European Journal of Operational Research*, *146*, 352–364.

Alter, S. (2008). Defining information systems as work systems: Implications for the IS field. *European Journal of Information Systems*, *17*(5), 448–469. https://doi.org/10.1057/ejis.2008.37

Alter, S. (2013). Work system theory: Overview of core concepts, extensions, and challenges for the future. *Journal of the Association for Information Systems*, *14*(2), 72–121. https://doi.org/10.17705/1jais.00323

Antonucci, Y. L., Corbitt, G., Stewart, G., & Harris, A. L. (2004). Enterprise systems education: Where are we? Where are we going? *Journal of Information Systems Education*, *15*(3), 227–234.

Appleget, D. J., Blais, M. C., Burks, C. R., Brown, C. R. F., Duong, D. D., Jaye, D. M., Perkins, M. T., & Thompson, M. (2010). *Irregular warfare (IW) model validation best practices guide* (p. 175) [Technical Report]. U.S. Army TRADOC Analysis Center.

Auer, D., Geist, V., & Draheim, D. (2009). *Extending BPMN with submit/response-style user interaction modeling*. IEEE Conference on Commerce and Enterprise Computing.

Avison, D. E., Wood-Harper, A. T., Vidgen, R. T., & Wood, J. R. G. (1998). A further exploration into information systems development: The evolution of multiview2. *Information Technology & People*, *11*(2), 124–139.

Axmann, B., Harmoko, H., Herm, L.-V., & Janiesch, C. (2021). A framework of cost drivers for robotic process automation projects. In J. González Enríquez, S. Debois, P. Fettke, P. Plebani, I. van de Weerd, & I. Weber (Eds.), *Business Process Management: Blockchain and Robotic Process Automation Forum* (pp. 7–22). Springer International Publishing.

Badr, N. M., Elabd, E., & Abdelkader, H. M. (2016). A semantic based framework for facilitating integration in ERP systems. *Proceedings of the 10th International Conference on Informatics and Systems*, 35–42.

Bailey, T. J., Dunlop, M. W., Mostow, J. R., Patterson, C., Tafoni, T., & Yeffeth, D. A. (2004). Aggregate level communications effects server: Meeting the communications realism challenge for large-scale real-time simulation experiments and analysis. *IEEE MILCOM 2004. Military Communications Conference, 2004.*, *2*, 867–873. https://doi.org/10.1109/MILCOM.2004.1494926

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Proceedings of the 1994 Winter Simulation Conference*, 215–220.

Bassellier, G., Reich, B. H., & Benbasat, I. (2001). Information technology competence of business managers: A definition and research model. *Journal of Management Information Systems*, *17*(4), 159–182.

Bisantz, A. M., & Vicente, K. J. (1994). Making the abstraction hierarchy concrete. *International Journal of Human-Computer Studies*, *40*, 83–117.

Blais, C. (2016). Challenges in representing human-robot teams in combat simulations. *Modelling and Simulation for Autonomous Systems* (pp. 3–16).

Blais, C. L. (2022, February). *Do we really want to solve interoperability?* [Workshop]. 2022 Simulation Innovation Workshop, Orlando, FL. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=53463&PortalId=0&TabId=105

Borgida, A., Mylopoulos, J., & Wong, H. K. T. (1984). Generalization/specialization as a basis for software specification. In M. L. Brodie, J. Mylopoulos, & J. W. Schmidt (Eds.), *On conceptual modeling* (pp. 87–117). Springer.

Bork, D. (2015). A development method for the conceptual design of multi-view modeling tools with an emphasis on consistency requirements [Dissertation]. University of Bamberg.

Bork, D., & Alter, S. (2020). Satisfying four requirements for more flexible modeling methods: Theory and test case. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, 3:1-25 Pages. https://doi.org/10.18417/EMISA.15.3

Bowen, J., Dittmar, A., & Weyers, B. (2021). Task modelling for interactive system design: A survey of historical trends, gaps and future needs. *Proceedings of the ACM on Human-Computer Interaction*, *5*(EICS), 1–22. https://doi.org/10.1145/3461736

Bowers, A., Budde, C. L., Gregg, B. C., & Winkowski, D. (2011). *JLVC data initialization*. [Workshop]. Fall SISO Simulation Interoperability Workshop 2011, Orlando, FL. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=34957&PortalId=0&TabId=105

Bram, S., & Vestergran, S. (2012). *Emergency response systems: Concepts, features, evaluation and design*. Center for Advanced Research in Emergency Response (CARER).

Brodie, M. L., Mylopoulos, J., & Schmidt, J. W. (Eds.). (1984). On conceptual modeling: Perspectives from artificial intelligence, databases, and programming languages. Springer-Verlag.

Bruseberg, A. (2008). Human views for MODAF as a bridge between human factors integration and systems engineering. *Journal of Cognitive Engineering and Decision Making*, *2*(3), 220–248. https://doi.org/10.1518/155534308X377090

Burland, B. R., Hyndøy, J. I., & Ruth, J. L. (2014, June). *Incorporating C2-simulation interoperability services into an operational command post*. 19th International Command and Control Research and Technology Symposium (ICCRTS), Alexandria, VA. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=42371&PortalId=0&TabId=105

Butler, K. A., Bahrami, A., Esposito, C., & Hebron, R. (2000). Conceptual models for coordinating the design of user work with the design of information systems. *Data & Knowledge Engineering*, *33*, 191–198.

Butler, K. A., Esposito, C., & Hebron, R. (1999). Connecting the design of software to the design of work. *Communications of the ACM*, *42*(1), 38–46. https://doi.org/10.1145/291469.293166

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, *15*(3), 289–308. https://doi.org/10.1016/S0953-5438(03)00010-9

Cannon-Bowers, J. A., Salas, E., & Grossman, J. D. (1992). Improving tactical decision making under stress: Research directions and applied implications. S. C. Collyer (Ed.), *The 27th International Applied Military Psychology Symposium: A Focus on Decision Making Research* (pp. 49–71).

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle. *ACM Transactions on Information Systems*, *10*(2), 181–212.

Caya, O., Leger, P.-M., Grebot, T., & Brunelle, E. (2014). Integrating, sharing, and sourcing knowledge in an ERP usage context. *Knowledge Management Research & Practice*, *12*(2), 193–202.

Charland, P., Leger, P.-M., Cronan, T., & Robert, J. (2015). Developing and assessing ERP competencies: basic and complex knowledge. *Journal of Computer Information Systems*, *56*(1), 31–39.

Charles, R., Sharples, S., Rajan, J. A., Wilson, J. R., & Wood, J. (2015). Analysing and designing control facilities. In J. R. Wilson and S. Sharples (Eds.), *Evaluation of human work* (4th Edition, pp. 383–415). CRC Press.

Cicchetti, A., Ciccozzi, F., & Leveque, T. (2012a). A hybrid approach for multi-view modeling. *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling - MPM '12*. Innsbruck, Austria.

Cicchetti, A., Ciccozzi, F., & Leveque, T. (2012b). Supporting incremental synchronization in hybrid multi-view modelling. In J. Kienzle (Ed.), *Models in Software Engineering* (Vol. 7167, pp. 89–103). Springer. https://doi.org/10.1007/978-3-642-29645-1_11

Citino, R. M. (1999). The path to blitzkrieg: Doctrine and training in the German Army, 1920–1939. Lynne Rienner Publishers.

Clark, T., & Jones, R. (1999). Organisational interoperability maturity model for C2. *Proceedings of the Command and Control Research and Technology Symposium (CCRTS)*, *29*.

Cole Engineering Services, Inc. (2018). *MTWS task order—Marine Corps development: system change request 17588 CLC2S study* [Technical Report]. Program Manager Training Systems, MARCORSYSCOM.

Constans, I. (2020, January 9). First to automation, NASA takes RPA agencywide. *GovLoop*. https://www.govloop.com/first-to-automation-nasa-takes-rpa-agencywide/

Cronan, T., Leger, P.-M., Robert, J., Babin, G., & Charland, P. (2012). Comparing objective measures and perceptions of cognitive learning in an ERP simulation game: a research note. *Simulation & Gaming*, *43*(4), 461–480.

Cundius, C., & Alt, R. (2013). *Real-time or near real-time? - Towards a real-time assessment model*. Thirty Fourth International Conference on Information Systems, Milan.

Cundius, C., & Alt, R. (2017). *A process-oriented model to business value—The case of real-time IT infrastructures*. Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2017.609

Daugherty, P. R., & Wilson, H. J. (2018). *Human + machine: Reimagining work in the age of AI*. Harvard Business Review Press.

Davenport, T. H., & Snabe, J. H. (2011, Spring). How fast and flexible do you want your information, really? *MIT Sloan Management Review*. https://sloanreview.mit.edu/article/how-fast-and-flexible-do-you-want-your-information-really/

Dehnert, J., & van der Aalst, W. M. P. (2004). Bridging the gap between business models and workflow specifications. *International Journal of Cooperative Information Systems*, *13*(3), 289–332.

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, *19*(4), 9–30. https://doi.org/10.1080/07421222.2003.11045748

Deranek, K., McLeod, A., & Schmidt, E. (2017). ERP simulation effects on knowledge and attitudes of experienced users. *Journal of Computer Information Systems*, *59*(4), 373–383. https://doi.org/10.1080/08874417.2017.1373610

Dijkman, R. M., Quartel, D. A. C., & van Sinderen, M. J. (2008). Consistency in multi-viewpoint design of enterprise information systems. *Information and Software Technology*, *50*(7–8), 737–752. https://doi.org/10.1016/j.infsof.2007.07.007

Duffy, L. (1995). Team decision-making biases: An information-processing perspective. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision making in action: Models and methods* (Second, pp. 346–359). Ablex Publishing Corporation.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. J. (2013). *Fundamentals of business process management*. Springer.

Eggleston, Ro. (2002). Cognitive systems engineering at twenty something: Where do we stand? In *Cognitive engineering in the military aviation environments: Avoiding cogminutia fragmentosa* (pp. 15–77). Human Systems Information Analysis Center, U.S. Department of Defense.

Ehrhart, D. L. S., & Bigbee, A. J. (1999). Co-evolving C2 organizational processes, decision support technology, and education/training: The role of evaluation in cognitive systems engineering. *Proceedings of the NATO Symposium on Modelling & Analysis of Command & Control*.

Elvesæter, B., Panfilenko, D., Jacobi, S., & Hahn, C. (2010). Aligning business and IT models in service-oriented architectures using BPMN and SoaML. *Proceedings of the First International Workshop on Model-Drive Interoperability - MDI '10*, 61–68. https://doi.org/10.1145/1866272.1866281

Engel, J., Herdin, C., & Märtin, C. (2014). Review of user interface description languages. *Proceedings of the 6th Forum Medientechnik*, 183–198.

Federal RPA Community of Practice. (2020). *RPA Program Playbook*.

Fengel, J. (Ed.). (2013). Business semantics alignment for business process model integration. In K. Tarnay, S. Imre, & L. Xu (Eds.), *Research and development in e-business through service-oriented solutions* (pp. 91–112). IGI Global. https://doi.org/10.4018/978-1-4666-4181-5

Fengel, J., Kohlborn, T., & Mueller, O. (2014). Semantic technologies for aligning heterogeneous business process models. *Business Process Management Journal*, *20*(4), 549–570.

Fengel, J., & Rebstock, M. (2010). Linking heterogenous conceptual models through a unifying modeling concepts ontology. *Proceedings of the 5th International Workshop on Semantic Business Process Management*, 1–4.

Fernandez, D., & Aman, A. (2018). Impacts of robotic process automation on global accounting services. *Asian Journal of Accounting and Governance*, *9*, 127–140.

Flach, J. M. (2013). Synthetic task environments and the three body problem. *Psychology Faculty Publications, CORE Scholar*. https://corescholar.libraries.wright.edu/psychology/269

Flach, J. M., & Dominguez, C. O. (1995). Use-centered design. *Ergonomics in Design*, 19–24.

Flach, J. M., & Hoffman, R. R. (2012). The limitations of limitations. In R. R. Hoffman, P. Hays, K. M. Ford, & J. M. Bradshaw (Eds.), *Collected essays on human-centered computing: 2001–2011* (pp. 171–174). IEEE Computer Society Press.

Flach, J. M., & Voorhorst, F. (2020a). *A meaning processing approach to cognition: What matters?* Routledge.

Flach, J. M., & Voorhorst, F. (2020b). Expertise: A holistic, experience-centered perspective. In P. Ward, J. M. Schraagen, J. Gore, & E. Roth (Eds.), *The Oxford handbook of expertise* (pp. 173–189). Oxford University Press.

Frerichs, M., Leible, S., & Nüttgens, M. (2021). Towards method- and tool-independent business process modeling. *INFORMATIK 2021*, 1567–1572.

Gable, G. G., Sedera, D., & Chan, T. (2008). Re-conceptualizing information system success: The IS-impact measurement model. *Journal of the Association for Information Systems*, *9*(7), 377–408.

Garcia, J. G. (2010). A methodology for developing user interfaces to workflow information systems [Dissertation]. University of Louvain.

Garcia, J. G., Vanderdonckt, J., & Calleros, J. M. G. (2008). FlowiXML: A step towards designing workflow management systems. *International Journal of Web Engineering and Technology*, *4*(2), 163. https://doi.org/10.1504/IJWET.2008.018096

Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, *13*(2), 209–228.

Gnewuch, U., Morana, S., Adam, M. T. P., & Maedche, A. (2018). *Faster is not always better: Understanding the effect of dynamic response delays in human-chatbot interaction*. Twenty-Sixth European Conference on Information Systems, Portsmouth, UK.

Gonzalez, R. A. (2009). Validation of crisis response simulation within the design science framework. International Conference on Information Systems. https://aisel.aisnet.org/icis2009/87/

Gonzalez, R. A. (2010). *A framework for ICT-supported coordination in crisis response* [Dissertation]. Delft University of Technology.

Gonzalez-Perez, C. (2017). How ontologies can help in software engineering. In J. Cunha, J. P. Fernandes, R. Lämmel, J. Saraiva, & V. Zaytsev (Eds.), *Grand Timely Topics in Software Engineering* (Vol. 10223, pp. 26–44). Springer International Publishing. https://doi.org/10.1007/978-3-319-60074-1_2

Guizzardi, G., Ferreira Pires, L., & van Sinderen, M. (2005). An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages. 691–705. https://doi.org/10.1007/11557432_51

Guizzardi, G., Pires, L. F., & van Sinderen, M. J. (2002). On the role of domain ontologies in the design of domain-specific visual modeling languages. *Proceedings of the 2nd Workshop on Domain-Specific Visual Languages*.

Gupta, S., Bostrom, R. P., & Huber, M. (2010). End-user training methods: What we know, need to know. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems*, *41*(4), 9–39.

Gurr, C. A. (1998). On the isomorphism, or lack of it, of representations. In K. Marriott and B. Meyer (Eds.), *Visual language theory* (pp. 293–306). Springer.

Hackathorn, R. (2004). Real-time to real-value. *DM Review, 14(1)*, 24–29. https://libproxy.nps.edu/login?url=https://www.proquest.com/scholarly-journals/real-time-value/docview/214681587/se-2

Hamill, J., Knight, F., Navarro, M., & Ressler, R. L. (2001). *Verification and validation (V&V) of C4ISR stimulation for testing and training*. [Workshop]. SISO Spring Simulation Interoperability Workshop 2001, Orlando. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=23522&PortalId=0&TabId=105

Hanley, J. (2017). Changing DOD's analysis paradigm: The science of war gaming and combat/campaign simulation. *Naval War College Review*, *70*(1), 64–103. https://digital-commons.usnwc.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1033&context=nwc-review

Hannay, J. E., & Kikke, Y. (2019). Structured crisis training with mixed-reality simulations. *Proceedings of the 16th ISCRAM Conference*.

Headquarters Marine Corps. (2019). *Commandant's planning guidance: 38th Commandant of the Marine Corps*. U.S. Marine Corps. https://www.hqmc.marines.mil/Portals/142/Docs/%2038th%20Commandant%27s%20Planning%20Guidance_2019.pdf?ver=2019-07-16-200152-700

Heffner, K., Mevassvik, O. M., Gautreau, B., & De Reus, N. (2014). A proposed process and toolset for developing standardized C2-to-simulation interoperability solutions. *Proceedings of the NATO Modelling and Simulation Group Symposium on Integrating Modelling & Simulation in the Defence Acquisition Life cycle and Military Training Curriculum (STO-MP-MSG-126)*.

Hensien, J. R. (2016). 1st Marine Logistics Group after action review for marine expeditionary force exercise / Large Scale Exercise 2016. 1st Marine Logistics Group.

Herzog, M. A., & Katzlinger-Felhofer, E. (2011). Influence of learning styles on the acceptance of game based learning in higher education: Experiences with a role playing simulation game. *Proceedings of the European Conference on Games-Based Learning*.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–100.

Hieb, M. R., & Timian, D. H. (1999). *Using army force-on-force simulations to stimulate C4I systems for testing and experimentation:* Defense Technical Information Center. https://doi.org/10.21236/ADA461500

Hindel, J., Cabrera, L. M., & Stierle, M. (2020, March). *Robotic process automation: Hype or hope?* 15th International Conference on Wirtschaftsinformatik, Potsdam, Germany.

Hoffman, R. R. (2012). Influencing versus informing design, Part 2: Macrocognitive modeling. In R. R. Hoffman, P. Hays, K. M. Ford, & J. M. Bradshaw (Eds.), *Collected essays on human-centered computing* (pp. 155–158). IEEE Computer Society Press.

Hoffman, R. R., Feltovich, P. J., Ford, K. M., Woods, D. D., Klein, G., & Feltovich, A. (2012). A rose by any other name...would probably be given an acronym. In R. R. Hoffman, P. Hays, K. M. Ford, & J. M. Bradshaw (Eds.), *Collected essays on human-centered computing, 2001–2011* (pp. 105–115). IEEE Computer Society Press.

Hoffman, R. R., & Fiore, S. M. (2012). Perceptual (re)learning: A leverage point for human-centered computing. In R. R. Hoffman, P. Hays, K. M. Ford, & J. M. Bradshaw (Eds.), *Collected essays on human-centered computing, 2001–2011* (pp. 205–211). IEEE Computer Society Press.

Hoffman, R. R., & Militello, L. G. (2009). Perspectives on cognitive task analysis: Historical origins and modern communities of practice. Psychology Press: Taylor & Francis Group.

Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2019). *Metrics for explainable AI: Challenges and prospects* (arXiv: 1812.04608; p. 50).

Hoffman, R. R., Ward, P., Feltovich, P. J., DiBello, L., Fiore, S. M., & Andrews, D. H. (2014). *Accelerated expertise: Training for high proficiency in a complex world*. Psychology Press.

Hofmann, M., Palii, J., & Mihelcic, G. (2011). Epistemic and normative aspects of ontologies in modelling and simulation. *Journal of Simulation*, *5*, 135–146.

Hollands, J. G., & Neyedli, H. (2011). A reliance model for automated combat identification systems: Implications for trust in automation. In N. Stanton (Ed.), *Trust in military teams* (pp. 151–181). CRC Press.

Hollnagel, E., & Woods, D. D. (2005). Joint cognitive systems: Foundations of cognitive systems engineering. CRC Press.

IEEE Std 1730–2010. (2011). IEEE recommended practice for distributed simulation engineering and execution process (DSEEP). IEEE Computer Society.

IEEE Std 1730.1-2013. (2013). IEEE recommended practice for distributed simulation engineering and execution process multi-architecture overlay (DMAO). IEEE Computer Society. https://doi.org/10.1109/IEEESTD.2013.6654219

Isoda, S. (2001). Object-oriented real-world modeling revisited. *Journal of Systems and Software*, *59*(2), 153–162. https://doi.org/10.1016/S0164-1212(01)00059-0

Jain, A., Fujimoto, R., Kim, J., Liu, M., Crittenden, J., & Lu, Z. (2015). Towards automating the development of federated distributed simulations for modeling sustainable urban infrastructures. *Proceedings of the 2015 Winter Simulation Conference*.

Jasperson, J., Carter, P. E., & Zmud, R. W. (2005). A comprehensive conceptualization of post-adoptive behaviors associated with information technology enabled work systems. *MIS Quarterly*, *29*(3), 525–557.

Jones, M. C. (2015). Composability. In L. B. Rainey and A. Tolk (Eds.), *Modeling and simulation support for system of systems engineering applications* (pp. 45–73). John Wiley & Sons, Inc.

Kackley, T. C. (2022, May). Future force modeling and simulation. *Marine Corps Gazette*, *106*(5), 53–55.

Kahneman, D. (2013). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Kahneman, D., & Klein, G. (2009). Conditions for intuitive expertise: A failure to disagree. *American Psychologist*, *64*(6), 515–526.

Kang, D., & Santhanam, R. (2003). A longitudinal field study of training practices in a collaborative application environment. *Journal of Management Information Systems*, *20*(3), 257–281.

Kaschek, R. (2008). On the evolution of conceptual modeling. In L. Delcambre, R. H. Kaschek, & H. C. Mayr (Eds.), *The Evolution of Conceptual Modeling* (Vol. 8181, pp. 1–12). Schloss Dagstuhl – Leiniz-Zentrum. https://drops.dagstuhl.de/opus/frontdoor.php?source_opus=1596

Kasparov, G. (2017). Deep thinking: Where machine intelligence ends and human creativity begins. Public Affairs.

Kim, K. (2015). Design of a framework to measure the degree of live virtual constructive (LVC) simulation interoperability [Dissertation]. University of Central Florida.

Kirk, R. E. (1996). Practical significance: A concept whose time has come. *Educational and Psychological Measurement*, *56*(5), 746–759.

Kirwan, B., & Ainsworth, L. K. (1992). *A guide to task analysis*. Taylor & Francis.

Klein, G. A. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision making in action: Models and methods* (pp. 138–147). Ablex Publishing Corporation.

Klein, G., Ross, K. G., Moon, B. M., Klein, D. E., Hoffman, R. R., & Hollnagel, E. (2003). Macrocognition. *IEEE Intelligent Systems*, 18(3). 81–84.

Klein, K. J., & Ralls, R. S. (1997). The unintended organizational consequences of technology training: Implications for training theory, research, and practice. In J. K. Ford, S. W. J. Kozlowski, K. Kraiger, E. Salas, & M. S. Teachout (Eds.), *Improving training effectiveness in work organizations* (pp. 323–354). Lawrence Erlbaum Associates.

Kolb, J., Reichert, M., & Weber, B. (2012). Using concurrent task trees for stakeholder-centered modeling and visualization of business processes. *S-BPM one - education and industrial developments*, 237–252.

Kovacevic, S. (1999). UML and user interface modeling. *The unified modeling language <<UML>> ‘98: Beyond the notation*, 253–266.

Kristiansen, R., & Trætteberg, H. (2007). Model-based user interface design in the context of workflow models. In M. Winckler, H. Johnson, & P. Palanque (Eds.), *Task models and diagrams for user interface design: TAMODIA 2007* (pp. 227–239). Springer. https://doi.org/10.1007/978-3-540-77222-4_18

Krogstie, J. (2012). *Model-based development and evolution of information systems.* Springer.

Krogstie, J., Dalberg, V., & Jensen, S. M. (2008). Process modeling value framework. *Enterprise Information Systems*, *3*, 309–321. https://doi.org/10.1007/978-3-540-77581-2_21

Lawrence, C. (2017). *War by numbers: Understanding conventional combat.* Potomac Books.

Lee, J. D. (2010). Demand calibration in multitask environments: Interactions of micro and macrocognition. In J. E. Miller and E. S. Patterson (Eds.), *Macrocognition metrics and scenarios: Design and evaluation of real-world teams* (pp. 95–108). CRC Press.

Lee, J., Lin, Y.-Y., Ma, S.-P., Wang, Y.-C., & Lee, S.-J. (2008). *Integrating service composition flow with user interactions*. IEEE International Symposium on Service-Oriented System Engineering. https://ieeexplore.ieee.org/document/4730471

Leger, P.-M., Charland, P., Feldstein, H. D., Robert, J., Babin, G., & Lyle, D. (2011). Business simulation training in information technology education: Guidelines for new approaches in IT training. *Journal of Information Technology Education*, *10*, 39–53.

LeMere, D. (2019). NASA Shared Services Center pioneers robotics process automation (RPA) for federal agencies. *IT Talk*. Office of the Chief Information Officer of the National Aeronautics and Space Administration. https://www.nasa.gov/sites/default/files/atoms/files/396062_jan-jun_2019_it_talk_design_final.pdf

Leopold, H., van der Aa, H., & Reijers, H. A. (2018). Identifying candidate tasks for robotic process automation in textual process descriptions. In J. Gulden, I. Reinhartz-Berger, R. Schmidt, S. Guerreiro, W. Guédria, & P. Bera (Eds.), *Enterprise, business-process and information systems modeling* (pp. 67–81). Springer International Publishing.

Leshob, A., Blal, R., Mili, H., Hadaya, P., & Hussain, O. K. (2019). From BPMN models to SoaML models. In L. Barolli, F. Hussain, & M. Ikeda (Eds.), *Complex, intelligent, and software intensive systems* (pp. 123–135). Springer.

Limbourg, Q. (2004). *Multi-path development of user interfaces* [Dissertation]. Catholic University of Louvain.

Limbourg, Q., & Vanderdonckt, J. (2004). Comparing task models for user interface design. In D. Diaper and N. A. Stanton (Eds.), *The handbook of task analysis for human-computer interaction* (6th ed., pp. 135–154). CRC Press.

Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., & Florins, M. (2004). *USIXML: A user interface description language supporting multiple levels of independence*. 325–338.

Liu, J., Yu, Y., Zhang, L., & Nie, C. (2011). An overview of conceptual model for simulation and its validation. *Procedia Engineering, 24*, 152–158.

Loffler, A., Levkovskyi, B., Prifti, L., Kienegger, H., & Krcmar, H. (2019, February). *Teaching the digital transformation of business processes: Design of a simulation game for information systems education*. 14th International Conference on Business Informatics, Siegen, Germany.

Lurie, N. H., & Swaminathan, J. M. (2009). Is timely information always better? The effect of feedback frequency on decision making. *Organizational Behavior and Human Decision Processes*, *108*(2), 315–329. https://doi.org/10.1016/j.obhdp.2008.05.005

March, S., & Smith, G. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*, 251–266.

Martinie, C., Palanque, P., & Winckler, M. (2011). Structuring and composition mechanisms to address scalability issues in task models. *Human-Computer Interaction - INTERACT 2011*, 589–609.

McGinnis, L., Huang, E., Kwon, K. S., & Uston, V. (2011). Ontologies and simulation: A practical approach. *Journal of Simulation*, *5*, 190–201.

McGrury, E. (2019, November 8). Getting the story right about wargaming. *War on the Rocks*. https://warontherocks.com/2019/11/getting-the-story-right-about-wargaming/

Miers, D., Kerremans, M., Ray, S., & Tornbohm, C. (2019). *Magic quadrant for robotic process automation software* (market analysis No. G00379618). Gartner. https://www.gartner.com/en/documents/3947184/magic-quadrant-for-robotic-process-automation-software

Mili, H., Jaoude, G. B., Lefebvre, É., Tremblay, G., & Petrenko, A. (2010). Business process modeling languages: Sorting through the alphabet soup. *ACM Computing Surveys*, *43*(4).

Militello, L. G., Dominguez, C. O., Lintern, G., & Klein, G. (2009). The role of cognitive systems engineering in the systems engineering design process. *Systems Engineering*, *13*(3), 261–273.

Miller, G.A. (1956). The magic number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81–97.

Miller, M. J., & Feigh, K. M. (2019). Addressing the envisioned world problem: A case study in human spaceflight operations. *Design Science*, *5*(3), 1–34.

Millet, P.-A., Schmitt, P., & Botta-Genoulaz, V. (2009). The SCOR model for the alignment of business processes and information systems. *Enterprise Information Systems*, *3*(4), 393–407.

Mineau, G. W., Missaoui, R., & Godinx, R. (2000). Conceptual modeling for data and knowledge management. *Data & Knowledge Engineering*, *33*, 137–168.

Morse, K. L., & Drake, D. L. (2021). Multi-viewpoint conceptual modeling in support of simulation interoperability readiness levels (SIRLs). NATO Modeling & Simulation Centre of Excellence.

Morse, K. L., Drake, D. L., Wells, D., & Bryan, D. (2014). *Realizing the cyber operational architecture training system (COATS) through standards*. [Workshop]. Fall Simulation Interoperability Workshop, Orlando.https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=42360&PortalId=0&TabId=105

Morse, M. M. (2016). The solution for LCE staff training. *Marine Corps Gazette*, *100*(1), 68–71.

Morse, M. M. (2017). Logistics staff trends. *Marine Corps Gazette*, *101*(10), 44–49.

Mudge, R. S., & Lingley, S. (2008). *Cyber and air joint effects demonstration (CAAJED)* (AFRL-RI-RS-2008-12). Air Force Research Laboratory.

Mulvaney, M. (2018). *Shifting from low-value to high-value work*. Office of Management and Budget. https://www.whitehouse.gov/wp-content/uploads/2018/08/M-18-23.pdf

Murray, B. & Curran, E. (2021, October 7). Beer distribution game: MIT students learn about supply chain. *Bloomberg*. https://www.bloomberg.com/news/articles/2021-10-07/beer-distribution-game-mit-sloan-students-learn-about-supply-chain

Murzek, M. (2008). The model morphing approach: Horizontal transformation of business process models [Dissertation]. Vienna University of Technology.

Mylopoulos, J. (1992). Conceptual modeling and telos. In P. Loucopoulos and R. Zicari (Eds.), *Conceptual modeling, databases, and CASE: An integrated view of information system development* (pp. 49–68). Wiley.

Mylopoulos, J. (1998). Information modeling in the time of the revolution. *Information Systems*, *23*(3–4).

NATO Science and Technology Organization. (2015a). *Simulation interoperability* (STO Technical Report TR-MSG-086-Part-I). NATO.

NATO Science and Technology Organization. (2015b). *Standardisation for C2-simulation interoperation*. (STO Technical Report TR-MSG-085). NATO.

Nisula, K. (2019). Holistic business learning environment: Bringing practice and integration to business education [Academic Dissertation]. Tampere University.

Nisula, K., & Pekkola, S. (2017). Business skills laboratory in action: Combining practice enterprise model and an ERP-simulation to a comprehensive business learning environment. *International Journal of Educational and Pedagogical Sciences*, *11*(10), Article 10.

Noppen, P., Beerepoot, I., van de Weerd, I., Jonker, M., & Reijers, H. A. (2020). How to keep RPA maintainable? In D. Fahland, C. Ghidini, J. Becker, & M. Dumas (Eds.), *Business process management* (pp. 453–470). Springer International Publishing.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner and A. L. Stevens (Eds.), *Mental models* (pp. 7–15). Lawrence Erlbaum Associates.

Oberle, D., Grimm, S., & Staab, S. (2009). An ontology for software. In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (pp. 383–402). Springer. https://doi.org/10.1007/978-3-540-92673-3_17

Office of the Inspector General. (2015). *FEMA Faces Challenges in Managing Information Technology* (OIG-16-10). U.S. Department of Homeland Security. https://www.oig.dhs.gov/reports/2016/fema-faces-challenges-managing-information-technology/oig-16-10-nov15

Osmundsen, K., Iden, J., & Bygstad, B. (2019, January 11). *Organizing robotic process automation: Balancing loose and tight coupling*. The 52nd Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2019.829

Pace, D. K. (2010). Conceptual modeling evolution within U.S. defense communities: The view from the Simulation Interoperability Workshop. In S. Robinson, R. Brooks, K. Kotiadis, & D. Van Der Zee (Eds.), *Conceptual Modeling for Discrete-Event Simulation* (pp. 439–466). Routledge.

Patterson, E. S., Roth, E. M., & Woods, D. D. (2010). Facets of complexity in situated work. In J. E. Miller and E. S. Patterson (Eds.), *Macrocognition metrics and scenarios: Design and evaluation for real-world teams* (pp. 221–251). CRC Press.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. *Proceeding of First International Conference on Design Science Research in Information Systems and Technology*.

Pennock, M. J., & Gaffney, C. (2018). Managing epistemic uncertainty for multimodels of sociotechnical systems for decision support. *IEEE Systems Journal*, *12*(1), 184–195. https://doi.org/10.1109/JSYST.2016.2598062

Perla, P. (2011). *The art of wargaming: A guide for professionals and hobbyists* (Second Edition). The History of Wargaming Project.

Perla, P., Ewell, W., Ma, C., Peach, J., Sepinsky, J., & Tripsas, B. (2019, October 21). Rolling the iron dice: From analytical wargaming to the cycle of research. *War on the Rocks*. https://warontherocks.com/2019/10/rolling-the-iron-dice-from-analytical-wargaming-to-the-cycle-of-research/

Pine, J. C. (2018). *Technology and Emergency Management (*Second Edition*)*. Wiley.

Pintus, A., Paterno, F., & Santoro, C. (2010). Modelling user interactions in web service-based business processes. *Proceedings of the 6th International Conference on Web Information Systems and Technology*, 175–180.

Polites, G. L. (2006). From real-time BI to the real-time enterprise: Organizational enablers of latency reduction. *International Conference on Information Systems 2006 Proceedings*, 1383–1400. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1206&context=icis2006

Pontico, F., Farenc, C., & Winckler, M. (2006). *Model-based support for specifying eService eGovernment applications*. [Workshop]. Task Models and Diagrams for Users Interface Design, 5th International Workshop, TAMODIA, Hasselt, Belgium.

Potter, S. S., & Rousseau, R. (2010). Evaluating the resilience of a human-computer decision-making team: A methodology for decision-centered testing. In J. E. Miller and E. S. Patterson (Eds.), *Macrocognition metrics and scenarios: Design and evaluation for real-world teams* (pp. 253–270). CRC Press.

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). *Strategies for design science research evaluation*. European Conference on Information Systems. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1214&context=ecis2008

Proctor, R. W., & Van Zandt, T. (2018). *Human factors in simple and complex systems* (3rd Edition). CRC Press.

Puerta, A., & Eisenstein, J. (1999). Towards a general computational framework for model-based interface development systems. *Knowledge-Based Systems*, *12*(8), 433–442.

Puerta-Melguizo, M. C., Chisalita, C., & van der Veer, G. C. (2002). Assessing users mental models in designing complex systems. *IEEE SMC*.

Pullen, J. M., & Khimeche, L. (2014, June). *Advances in systems and technologies toward interoperating operational military C2 and simulation systems*. 19th International Command and Control Research and Technology Symposium (ICCRTS), Alexandria, VA.

Pullen, J. M., & Mevassvik, O. M. (2016). Coalition command and control – Simulation interoperation as a system of systems. *11th System of Systems Engineering Conference (SoSE)*, 1–6.

Pullen, J. M., & Ruth, J. (2018a). *Military training in a cyber-active environment exploiting C2-simulation interoperation*. [Workshop]. SISO Fall Simulation Innovation Workshop 2018, Orlando, FL. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=47982&PortalId=0&TabId=105

Pullen, J. M., & Ruth, J. (2018b). Training operational military organizations in a cyber-active environment using C2-simulation interoperation. 23rd ICCRTS.

Pütz, C., & Sinz, E. J. (2010). Model-driven derivation of BPMN workflow schemata from SOM business process models. *Enterprise Modelling and Information Systems Architectures*, *5*(2), 57–72.

Ressler, R. L., Hieb, M. R., & Sudnikovich, W. (1999). *M&S/C4ISR conceptual reference model*. [Workshop]. SISO Simulation Interoperability Workshop 1999, Orlando, FL. https://www.sisostds.org/Default.aspx?tabid=105&EntryId=21183

Reyes, P. M., Li, S., & Visich, J. K. (2016). Determinants of RFID adoption stage and perceived benefits. *European Journal of Operational Research*, *254*(3), 801–812. https://doi.org/10.1016/j.ejor.2016.03.051

Robinson, S. (2008). Conceptual modeling for simulation part I: Definition and requirements. *Journal of the Operational Research Society*, *59*(3), 278–290.

Roth, E. M., Lin, L., Kerch, S., Kenney, S. J., & Sugibayashi, N. (2001). Designing a first-of-a-kind group view display for team decision making: A case study. In E. Salas and G. Klein (Eds.), *Linking expertise and naturalistic decision making* (pp. 113–135). Lawrence Erlbaum Associates.

Roth, E. M., Patterson, E. S., & Mumaw, R. J. (2000). Cognitive engineering: Issues in user-centered systems. In J. J. Marciniak (Ed.), *Encyclopedia of software engineering* (2nd Edition). John Wiley & Sons, Inc.

Rowland, D. (2019). *The stress of battle: Quantifying human performance in battle for historical analysis and wargaming* (Second Edition). The History of Wargaming Project.

Rukhin, A., Soto, J., Nechvatal, J., Barker, E., Leigh, S., Levenson, M., Banks, D., Heckert, A., & Dray, J. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (Special publication 800–22; p. 131). National Institute of Standards and Technology.

Ryabko, B. Ya., Stognienko, V. S., & Shokin, Yu. I. (2004). A new test for randomness and its application to some cryptographic problems. *Journal of Statistical Planning and Inference*, *123*(2), 365–376. https://doi.org/10.1016/S0378-3758(03)00149-6

Rybacki, M. G., & Blackman, M. (1997). *Evolving a logistics modeling and simulation*. [Workshop]. Fall SISO Simulation Interoperability Workshop 1997, Orlando, FL. https://www.sisostds.org/Default.aspx?tabid=105&EntryId=19155

Salas, E., Priest, H., Wilson, K., & Burke, C. S. (2006). Scenario-based training: Improving military performance and adaptability. In T. W. Britt, A. B. Adler, & C. A. Andrew (Eds.), *Military life: The psychology of serving in peace and combat* (Vol. 1–2, pp. 32–53). Praeger Security International.

Salas, E., Wildman, J. L., & Piccolo, R. F. (2009). Using simulation-based training to enhance management education. *Academy of Management Learning & Education*, *8*(4), 559–573.

Salmon, P., Stanton, N. A., Walker, G. H., & Read, G. J. M. (2020). Studying expert behavior in sociotechnical systems. In P. Ward, J. M. Schraagen, J. Gore, & E. Roth (Eds.), *The Oxford handbook of expertise* (pp. 354–376). Oxford University Press.

Salmon, P., Stanton, N., Jenkins, D., Walker, G., & Rafferty, L. (2010). Decisions, decisions...and even more decisions: Evaluation of a digitized mission support system in the land warfare domain. *International Journal of Human-Computer Interaction*, *26*, 206–227.

Sanderson, P. (2018). Understanding cognitive work. In P. J. Smith and R. R. Hoffman (Eds.), *Cognitive systems engineering: The future of a changing world*. CRC Press.

Sandkuhl, K., Fill, H.-G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö., & Winter, R. (2018). From expert discipline to common practice: A vision and research agenda for extending the reach of enterprise modeling. *Business & Information Systems Engineering*, *60*(1), 69–80. https://doi.org/10.1007/s12599-017-0516-y

Sargent, R. (2007). Verification and validation of simulation models. *Proceedings of the 2007 Winter Simulation Conference*, 124–137. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4419595

Sein, M., Bolstrom, R. P., & Olfman, L. (1999). Rethinking end-user training strategy: Applying a hierarchical knowledge-level model. *Journal of Organizational and End User Computing (JOEUC)*, *11*(1), 32–39.

Shattuck, L. G., & Miller, N. L. (2006). Extending naturalistic decision making to complex organizations: A dynamic model of situated cognition. *Organization Studies*, *27*(7).

Shattuck, L., Talcott, C., Matthews, M., Clark, J., & Swiergosz, M. (2002). Constructing battlefield understanding: A comparison of experienced and novice decision makers in different contexts. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *46*, 443–447.

Sheehan, J., Prosser, T., Conley, H., Stone, G., Yentz, K., & Morrow, J. (1998). Conceptual models of the mission space (CMMS): Basic concepts, advanced techniques, and pragmatic examples. *Proceedings of Spring Simulation Interoperability Workshop*.

Shen, A. (2020). Randomness tests: Theory and practice. In A. Blass, P. Cégielski, N. Dershowitz, M. Droste, & B. Finkbeiner (Eds.), *Fields of Logic and Computation III* (Vol. 12180, pp. 258–290). Springer International Publishing. https://doi.org/10.1007/978-3-030-48006-6_18

Simon, H. A. (1996). *The sciences of the artificial* (3rd Edition). MIT Press.

SISO. (2022, May 9). *SISO verification, validation & accreditation/acceptance (VV&A) products PDG*. https://www.sisostds.org/StandardsActivities/DevelopmentGroups/VVAProductsPDG.aspx

SISO FEAT Product Support Group. (2017, January 23). *Federation engineering agreements template (FEAT) user's guide*. (SISO-REF-067-2017).

SISO Standards Activity Committee. (2020). Final report of the Cyber Modeling and Simulation (M&S) Study Group (SG) (SISO-REF-070-2020).

SISO-STD-012-2013. (2013). *Standard for federation engineering agreements template (FEAT).* SISO. https://www.sisostds.org/featprogrammersreference/index.htm

Smith, P. J., & Hoffman, R. R. (Eds.). (2018). Cognitive systems engineering: The future for a changing world. CRC Press.

Sousa, K., Mendonca, H., & Vanderdonckt, J. (2009). User interface development life cycle for business-driven enterprise applications. In V. L. Jaquero, F. M. Simarro, J. P. M. Masso, & J. Vanderdonckt (Eds.), *Computer-aided design of user interfaces VI* (pp. 23–34). Springer.

Sousa, K., Mendonça, H., & Vanderdonckt, J. (2010). A rule-based approach for model management in a user interface – Business alignment framework. In D. England, P. Palanque, J. Vanderdonckt, & P. J. Wild (Eds.), *Task models and diagrams for user interface design 2009* (pp. 1–14). Springer. https://doi.org/10.1007/978-3-642-11797-8_1

Sousa, K., & Vanderdonckt, J. (2011). Business performer-centered design of user interfaces. In H. Hussmann, G. Meixner, & D. Zuehlke (Eds.), *Model-driven development of advanced user interfaces* (pp. 123–142). Springer.

Stavness, N., & Schneider, K. A. (2004). Supporting flexible business processes with a progression model. Proceedings of the 1st International Workshop on Making Model-Based User Interface Design Practical: Usable and Open Methods and Tools MBUI 2004.

Stein, S., Kuhne, S., & Ivanov, K. (2009). *Business to IT transformations revisited*. *17*, 176–187.

Stuetelberg, M. B., & Thomas, J. R. (2021). Incorporating predictive maintenance best practices into Marine Corps training and operations [Master's thesis]. Naval Postgraduate School. https://calhoun.nps.edu/handle/10945/68794

Tan, K. H., Tse, Y. K., & Chung, P. L. (2010). A plug and play pathway approach for operations management games development. *Computers & Education*, *55*, 109–117.

Tolk, A. (2006, September). *Next generation data interoperability: It's all about the metadata*. [Workshop]. Fall SISO Simulation Interoperability Workshop 2006, Orlando, FL. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=27138&PortalId=0&TabId=105

Tolk, A. (2012). Modeling sensing. In A. Tolk (Ed.), *Engineering principles of combat modeling and distributed simulation* (pp. 127–144). John Wiley & Sons, Inc.

Tolk, A. (2017). Interoperability and composability: A journey through mathematics, computer science, and epistemology. *Proceedings of NATO MSG Symposium, NATO Report STO-MP-MSG-149*.

Tolk, A., Diallo, S. Y., Padilla, J. J., & Turnitsa, C. (2012). How is M&S interoperability different from other interoperability domains? *M&S Journal*, *7*(3), 5–14.

Tolk, A., & Miller, J. A. (2011). Enhancing simulation composability and interoperability using conceptual/semantic/ontological models. *Journal of Simulation*, *5*, 133–134.

Tornbohm, C., & Dunie, R. (2017). *Market guide for robotic process automation software* (market analysis No. G00319864). Gartner. http://images.abbyy.com/India/market_guide_for_robotic_pro_319864%20(002).pdf

Trætteberg, H. (1999). Modeling work: Workflow and task modeling. Proceedings of the Third International Conference of Computer-Aided Design of User Interfaces, 275–280.

Trætteberg, H., & Krogstie, J. (2008). Enhancing the usability of BPM-solutions by combining process and user-interface modelling. *PoEM 2008*.

Training and Education Capabilities Division, Training and Education Command, USMC. (2018). *MAGTF-TITE working group technical AAR for the USMC LVC-TE*. [Technical Report]. U.S. Marine Corps.

Turnitsa, C., Padilla, J. J., & Tolk, A. (2010). Ontology for modeling and simulation. *Proceedings of the 2010 Winter Simulation Conference*.

Ulmer, J.-S., Belaud, J.-P., & Le Lann, J.-M. (2013). A pivotal-based approach for enterprise business process and IS integration. *Enterprise Information Systems*, *7*(1), 61–78. https://doi.org/10.1080/17517575.2012.700326

United States Army Combined Arms Support Command. (2016a). *Sustainment constructive training white paper*.

United States Army Combined Arms Support Command. (2016b). *Sustainment training strategy and guide*.

USD(AT&L). (2009). DOD modeling and simulation (M&S) verification, validation, and accreditation (VV&A) (DoDI 5000.61). U.S. Department of Defense.

Using Innovative Technology and Practices to Enhance the Culture of Preparedness, Subcommittee on Emergency Preparedness, Response, and Communications, U.S. House Committee on Homeland Security, 114th Cong. (2018) (testimony of John V. Kelly). https://docs.house.gov/meetings/HM/HM12/20180725/108605/HHRG-115-HM12-Wstate-KellyJ-20180725.pdf

Van Den Berg, T., & Lutz, R. (2015). *Simulation environment architecture development using the DoDAF*. [Workshop]. SISO Simulation Interoperability Workshop 2015, Orlando, FL. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=43325&PortalId=0&TabId=105

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business Information System Engineering*, *60*(4), 269–272.

van der Veer, G. C., & Puerta Melguizo, M. del C. (2002). Mental models. In A. Sears and J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 52–80). Lawrence Erlbaum Associates Inc.

Van Wyk, E., & Heimdahl, M. P. E. (2009). Flexibility in modeling languages and tools: A call to arms. *International Journal on Software Tools for Technology Transfer*, *11*(3), 203–215. https://doi.org/10.1007/s10009-009-0107-4

Vanderdonckt, J. (2005). A MDA-compliant environment for developing user interfaces of information systems. In R. King (Ed.), *Active Flow and Combustion Control 2018* (Vol. 141, pp. 16–31). Springer International Publishing. https://doi.org/10.1007/11431855_2

Venable, J. (2006). A framework for design science research activities. *Proceedings of the 2006 Information Resource Management Association Conference*, 184–187.

Venkatesan, D. (2018). A novel agent-based enterprise level system development technology [Dissertation]. Anna University.

Venkatesan, D., & Sundaramurthy, S. (2018). Promoting business-IT alignment through agent metaphor-based software technology. *International Journal of Information Technology and Management*, *10*.

Villalón-Huerta, A., Ripoll-Ripoll, I., & Marco-Gisbert, H. (2021). CNA tactics and techniques: A structure proposal. *Journal of Sensor and Actuator Networks*, *10*(1).

Völker, M., & Weske, M. (2021). Conceptualizing bots in robotic process automation. In A. Ghose, J. Horkoff, V. E. Silva Souza, J. Parsons, & J. Evermann (Eds.), *Conceptual modeling* (pp. 3–13). Springer International Publishing.

Walter, T., & Ebert, J. (2009). Combining DSLs and ontologies using metamodel integration. 148–169.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling—A research agenda. *Information Systems Research*, *13*(4), 363–376.

Wells, D., & Bryan, D. (2015). *Cyber operational architecture training system – Cyber for all*. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).

West, R. L., & Nagy, G. (2007). Using GOMS for modeling routine tasks within complex sociotechnical systems: Connecting macrocognitive models to microcognition. *Journal of Cognitive Engineering and Decision Making*, *1*(2), 186–211. https://doi.org/10.1518/155534307X232848

Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. (2016). *Engineering psychology and human performance* (Fourth Edition). Routledge.

Wieringa, R. J. (2014). Design science methodology for information systems and software engineering. Springer.

Wieringa, R., & Morali, A. (2012). Technical action research as a validation method in information systems design science. *Design science research in information systems: Advances in theory and practice*, 220–238.

Wilson, B. (2017). *Interfacing force-on-force and communications models: MANA and JNE.* RAND Corporation.

Wong, Y., Bae, S., Bartels, E., & Smith, B. (2019). *Next-generation wargaming for the U.S. Marine Corps: Recommended courses of action*. RAND Corporation. https://www.rand.org/pubs/research_reports/RR2227.html

Woods, D. D. (2002). Steering the reverberations of technology change on fields of practice: Laws that govern cognitive work. *Proceedings of the 24th Annual Conference of the Cognitive Science Society*.

Woods, D. D., & Hollnagel, E. (2006). Joint cognitive systems: Patterns in cognitive systems engineering. CRC Press.

Woods, D. D., & Roth, E. M. (1988). Cognitive systems engineering. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 3–43). Elsevier Science Publishers.

Woods, D., & Dekker, S. (2000). Anticipating the effects of technological change: A new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science*, *1*(32), 272–282.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California