An Agile Roadmap for Live, Virtual and Constructive-Integrating Training Architecture (LVC-ITA): A Case Study Using a Component based Integrated Simulation Engine

Tae Woong Park
University of Central Florida

AN AGILE ROADMAP FOR LIVE, VIRTUAL AND CONSTRUCTIVE-INTEGRATING TRAINING ARCHITECTURE (LVC-ITA): A CASE STUDY USING A COMPONENT BASED INTEGRATED SIMULATION ENGINE

by

TAE WOONG PARK
B.S. Korea Military Academy, 2000
M.S. Seoul National University, 2005
M.S. University of Central Florida, 2014

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2015

Major Professor: Gene Lee

ABSTRACT

Conducting seamless Live Virtual Constructive (LVC) simulation remains the most challenging issue of Modeling and Simulation (M&S). There is a lack of interoperability, limited reuse and loose integration between the Live, Virtual and/or Constructive assets across multiple Standard Simulation Architectures (SSAs). There have been various theoretical research endeavors about solving these problems but their solutions resulted in complex and inflexible integration, long user-usage time and high cost for LVC simulation.

The goal of this research is to provide an *Agile Roadmap for the Live Virtual Constructive-Integrating Training Architecture (LVC-ITA)* that will address the above problems and introduce interoperable LVC simulation. Therefore, this research describes how the newest M&S technologies can be utilized for LVC simulation interoperability and integration. Then, we will examine the optimal procedure to develop an agile roadmap for the LVC-ITA.

In addition, this research illustrated a case study using an Adaptive distributed parallel Simulation environment for Interoperable and reusable Model (AddSIM) that is a component based integrated simulation engine. The agile roadmap of the LVC-ITA that reflects the lessons learned from the case study will contribute to guide M&S communities to an efficient path to increase interaction of M&S simulation across systems.

Keywords: interoperability, integration, Modeling and Simulation (M&S), Live Virtual Constructive (LVC), Live, Virtual, Constructive-Integrating Training Architecture (LVC-ITA), Standard Simulation Architecture (SSA), Roadmap, AddSIM

ACKNOWLEDGMENTS

It is my pleasure now to express my gratefulness to all the people who were with me during this journey of learning. I would like to express my gratitude to my main advisor Dr. Gene Lee and co-advisor Dr. Luis Rabelo for all of their strong support, kindness, understanding and encouragement throughout my Ph.D. study at the University of Central Florida (UCF). I am also grateful to other doctoral committee members, Dr. Peter Kincaid and Dr. Ahmad Elshennawy for their thoughtful review and supportive advice.

The project was supported by the Agency for Defense Development (ADD), which helped me finish my dissertation and to continue the research of the roadmap on the LVC-ITA. In addition, Department of Industrial Engineering and Management Systems (IEMS) provided initial support and helped me to execute LVC simulation.

I thank members in the Simulation Interoperability Laboratory (SIL). Major. Kiyoul Kim who is Republic of Korea Air Force officer and my coworker also answered my questions related to the field of Modeling & Simulation (M&S) and gave me useful feedback to my practice talk at the lab. He also gave useful comments and encouragement for my dissertation and helped to execute the LVC simulation case study for LVC-ITA. In addition, Seong K. Lee who is my Christian friend helped me to improve my slides for the final defense and my writing of the dissertation.

Finally, I greatly thank my wife, Mi Ri Kim, for giving me abundant love and support during these four years. Now, I want to enjoy the sunny Florida weather with my adorable six-year-old daughter, Seoyeong Park, one-year-old daughter Seoyoon Katelyn Park and my wife, in Orlando, Florida.

TABLE OF CONTENTS

LIST OF FIGURES xiii

LIST OF TABLES xviii

LIST OF ACRONYMS (or) ABBREVIATIONS xx

CHAPTER ONE: INTRODUCTION 1

1.1 General Background 3

1.1.1 Stand-alone Simulation and Distributed Simulation... 3

1.1.2 Concept of Live, Virtual or Constructive Simulations 5

1.1.3 Concept of Live Virtual Constructive (LVC) Simulations 9

1.1.4 Overview of Standard Simulation Architecture (SSA)... 11

1.1.5 Overview of the U.S. Army Live, Virtual, Constructive-Integrating Architecture (LVC-IA) 13

1.2 Statement of the Problems 15

1.2.1 Problem 1: Inherent Limited Interoperability between the Different SSAs. 16

1.2.2 Problem 2: Many Issues in Integrating LVC Assets. 17

1.2.3 Problem 3: Decentralized Development of SSAs and LVC Assets............................ 18

1.3 Purpose, Goal and Objectives of the Research 19

1.3.1 Objective 1: Assessment of the Current State in an LVC Simulation Environment. .. 20

1.3.2 Objective 2: Making the Right Vision for the Future M&S 21

1.3.3 Objective 3: Conducting a case study reflecting current LVC simulation situation.... 21

1.3.4 Objective 4: Drawing lessons learned from the case study. 22

1.4 Expected Contributions.. 22

1.4.1 Contribution 1: Simpler Integration 23

1.4.2 Contribution 2: Flexible Integration 23

1.4.3 Contribution 3: Reuse Legacy Simulation Systems 24

1.4.4 Contribution 4: Cost-effective and Shorter Time to LVC User.................................. 24

1.5 Dissertation Outline 24

CHAPTER TWO: LITERATURE REVIEW 27

2.1 Efforts for Improving LVC Interoperability 27

2.1.1 Department of Defense (DoD) Modeling and Simulation (M&S) Master Plan 27

2.1.2 Joint Live Virtual Constructive Data Translator (JLVCDT) Framework.................... 28

2.2 Interoperability, Integration and Composability.. 29

2.2.1 Interoperability.. 30

2.2.2 Integration 36

2.2.3 Composability 37

2.3 Comparison of Standard Simulation Architectures (SSAs).................................. 37

2.3.1 SIMulation NETworking (SIMNET).. 38

2.3.2 Aggregate Level Simulation Protocol (ALSP) 38

2.3.3 Distributed Interactive Simulation (DIS) 39
2.3.4 High Level Architecture (HLA) 40
2.3.5 Test and Training Enabling Architecture (TENA) 44
2.3.6 Common Training Instrumentation Architecture (CTIA) 46
2.3.7 Comparison of SSAs 47
2.3.8 Section Summary 48
2.4 Conceptual Model (CM) 49
2.4.1 Modeling and Simulation (M&S) Process 50
2.4.2 Development of Conceptual Model of the CMMS and DCMF 54
2.4.3 Section Summary 56
2.5 Bridging Solutions 57
2.5.1 Gateway 57
2.5.2 Middleware 58
2.5.3 Broker 59
2.5.4 Proxy 60
2.5.5 Protocol Solution 60
2.5.6 Section Summary 60
2.6 U.S. DoD Live-Virtual-Constructive Architecture Roadmap (LVCAR) 60
2.6.1 Purpose of the LVCAR 61

2.6.2 Main Four Fundamental Precepts of the LVCAR 61

2.6.3 Section Summary 63

2.7 U.S. Army Live Virtual Constructive-Integrating Architecture (LVC-IA)........................ 63

2.7.1 Overview of LVC-IA 65

2.7.2 Training Case based on LVC Simulation 66

2.7.3 Training Concept of the U.S. Army LVC-IA 69

2.7.4 U.S. Army LVC-IA Capabilities 71

2.7.5 Components of the U.S. Army LVC-IA 73

2.7.6 Goal of the U.S. Army LVC-IA.. 103

2.7.7 Section Summary 104

2.8 Common Standards, Products, Architectures and/or Repositories (CSPAR).................. 104

2.9 Research Gap 105

2.9.1 Complex Integration 105

2.9.2 Long Time-to-LVC User-Usage 106

2.9.3 High Cost 107

2.9.4 Inflexible Integration 107

CHAPTER THREE: METHODOLOGY 108

3.1 Flow Chart of Methodology 108

3.2 Description of Methodology... 109

3.2.1 Step 1: Formal Problem Definition 109

3.2.2 Step 2: Literature Review 109

3.2.3 Step 3: Research Gap Analysis 110

3.2.4 Step 4: Design Requirements for a Case Study 110

3.2.5 Step 5: LVC Simulation Case Study 111

3.2.6 Step 6: Agile Roadmap for LVC-ITA 116

CHAPTER FOUR: CASE STUDY 117

4.1 Background 117

4.2 Planning a Case Study 118

4.3 Phase 1: Research Questions 118

4.4 Phase 2: Designing a Case Study 119

4.4.1 Component-based simulation environment: Adaptive distributed parallel simulation environment based on interoperable and reusable models (AddSIM) 121

4.4.2 Virtual Simulator: SIMbox 130

4.4.3 Constrictive Simulation System: VR-Forces 133

4.4.4 WebLVC Server 134

4.4.5 Data Logger 135

4.5 Phase 3: Conducting a Case Study 135

4.5.1 Objective 135

4.5.2 Member Applications 135
4.5.3 Prerequisite Condition 139
4.5.4 Designing Air Defense Engagement Scenario 139
4.5.5 Procedure of Air Defense Engagement Simulation 141
4.5.6 Simulation Result Analysis 146
4.6 Phase 4: Case Study Findings 147
4.6.1 Problem 1: Lack of Interactions between Simulation Entities 148
4.6.2 Problem 2: Lack of Reusability 150
4.6.3 Problem 3: Lack of Scalability and Interoperability of HLA federation 154
4.6.4 Problem 4: Limited Capability of Computer Generated Forces (CGFs) (or Semi-Automated Forces (SAFs)) 160
4.6.5 Problem 5: Limited Reference Models in Database 162
4.6.6 Problem 6: Limited Correlated Terrain Databases (TDBs) Representation 162
4.6.7 Problem 7: Limited Use of the Simulation Systems for Multipurpose 169
4.6.8 Problem 8: Limited Analysis of Engagement Result 170
4.7 Phase 5: Case Study Lessons Learned 176
4.7.1 Lesson Learned 1: Need for a Common Standard Simulation Entity 177
4.7.2 Lesson Learned 2: Need for an Entity Level Simulation Systems 178
4.7.3 Lesson Learned 3: Need for a Common Standard-Defense Conceptual Modeling. 181

4.7.4 Lesson Learned 4: Need for Computer Generated Forces (CGF) (or Semi-Automated Forces (SAFs)) 183

4.7.5 Lesson Learned 5: Need for Multiple SSAs Compliancy 186

4.7.6 Lesson Learned 6: Need for Scalability Capability of Simulation Systems 188

4.7.7 Lesson Learned 7: Need for a Common Correlated Terrain Databases (TDBs) 189

4.7.8 Lesson Learned 8: Need for a New Common Standard Simulation Architecture (C-SSA) 196

4.7.9 Lesson Learned 9: Need for a Product Line Architecture Framework (PLAF) Concept 196

4.7.10 Lesson Learned 10: Need for a Simulation System to Support Multiple M&S Applications and LVC simulations 198

4.7.11 Lesson Learned 11: Need for a Battle Damage Assessment (BDA) Application 199

4.7.12 Lesson Learned 12: Need for a General Bridging Tool 202

4.8 Phase 6: Recommended Actions 205

CHAPTER FIVE: AGILE ROADMAP FOR LVC-ITA 208

5.1 Recommended Action 1: Common Standard - Defense Modeling and Simulation Process (CS-DMSP) 210

5.1.1 Common Standard-Defense Modeling and Simulation Process (CS-DMSP) 210

5.1.2 Common Model 214

5.2 Recommended Action 2: Common Standard-Simulation System Architecture Framework (CS-SSAF) 218

5.2.1 Common Standard - Simulation System Architecture Framework (CS-SSAF) 219

5.2.2 Common Standard-Live Simulation System Architecture Framework (CS-LSSAF) 235

5.2.3 Common Standard-Virtual Simulation System Architecture Framework (CS-VSSAF) 240

5.2.4 Common Standard-Constructive Simulation System Architecture Framework (CS-CSSAF). 245

5.3 Recommended Action 3: Common Standard Correlated Terrain Database (CS-CTDB) 247

5.3.1 Strategy 1: Reuse Legacy Simulation System 248

5.3.2 Strategy 2: Develop CS-CTDB Generation System 250

5.4 Recommended Action 4: Advanced Interoperability Technology 253

5.4.1 Policy Establishment on Multiple Standard Simulation Architecture (SSAs) 254

5.4.2 Linking Strategy between CS-LSSAF, CS-VSSAF and CS-CSSAF 256

CHAPTER SIX: CONCLUSION 261

6.1 Summary 261

6.2 Contribution 263

6.3 Limitations and Future Work 266

LIST OF REFERENCES 268

LIST OF FIGURES

Figure 1: Stand-alone simulation. 3

Figure 2: Distributed simulation. 4

Figure 3: Live simulation: U.S. Air Force F-22 Raptor 6

Figure 4: Virtual simulation: F-16 Mission Training Center (MTC) 7

Figure 5: Constructive simulation 8

Figure 6: Graphic of an LVC Synthetic Environment 9

Figure 7: Historical Evolution of Standard Simulation Architecture (SSA) 13

Figure 8: LVC-IA 14

Figure 9: Overview of the Dissertation 26

Figure 10: DoD M&S Objective and Sub-Objective 28

Figure 11: Levels of Conceptual Interoperability Model 33

Figure 12: RTI and applications in the HLA 42

Figure 13: TENA Architecture 45

Figure 14: Usage Frequency of SSAs in the U.S. 47

Figure 15: Top level process of the FEDEP 51

Figure 16: Top level process of the SEDEP 52

Figure 17: Top level process of DSEEP 52

Figure 18: Four phases of the DCMF process 55

Figure 19: Gateway Configuration 58

Figure 20: Middleware Configuration 59

Figure 21: Brokers 59

Figure 22: LVCTE Objective Systems 65

Figure 23: JCATS workstation and Display 68

Figure 24: Close Combat Tactical Trainer (CCTT).. 68

Figure 25: California CCTT Mobile Units 69

Figure 26: LVC-IA Operational View.. 70

Figure 27: Three Major Components of the LVC-IA.. 74

Figure 28: LT2 Component Product Line Framework 75

Figure 29: CTIA Layered View... 79

Figure 30: LT2 FTS Operational View.. 82

Figure 31: Functional Breakdown of SE Core Program..................................... 83

Figure 32: SE Core Operational View within LVC Training 83

Figure 33: VSA Domain Context 85

Figure 34: PLAS Document Breakdown 86

Figure 35: VSA PLAF 87

Figure 36: Overall STDGC Process Concept 93

Figure 37: Detailed STDGC Process... 93

Figure 38: JLCCTC Objective Architecture.. 95

Figure 39: JLCCTC MRF V3 Architecture 96

Figure 40: WARSIM 3-Layer Architecture.. 98

Figure 41: WARSIM Abstract System Architecture 100

Figure 42: JLCCTC ERF V3 Logical Block Diagram...................................... 101

Figure 43: OneSAF Product Line Architecture Framework............................ 103

Figure 44: Common LVC Architecture Vision 106

Figure 45: Flow Chart of Methodology 108

Figure 46: Scenario Concept of LVC Simulation Case Study....................................... 112

Figure 47: Case Study Process... 118

Figure 48: HLA Federation for Air Defense Engagement... 120

Figure 49: HLA Federation with WebLVC for Air Defense Engagement..................... 120

Figure 50: Final Design for the LVC simulation case study 121

Figure 51: Layered Architecture of AddSIM... 123

Figure 52: A hierarchical modeling structure of AddSIM... 127

Figure 53: Operational concept of distributed repository. 129

Figure 54: F-16 Flight Simulator 131

Figure 55: SA-8 SAM Simulator 132

Figure 56: Plyers (or Entities) in the case study 136

Figure 57: Hardware and Software Specification of the Case Study Environment........ 136

Figure 58: Air Defense Engagement Scenario... 140

Figure 59: Geographical Condition and Initial Scenario Setting on VR-Forces GUI 141

Figure 60: Initial Situation of SA-8 SAM (Blue force) and F-16 Flight (Red force)..... 142

Figure 61: Join of AddSIM's Air Defense Radar Player from Initial Situation 143

Figure 62: F-16 Flight Information in AddSIM... 143

Figure 63: F-16 Flight Information in VR-Forces 144

Figure 64: SA-8 SAM's Attack to F-16 flight... 145

Figure 65: Data Logger's Record 146

Figure 66: Entities Mapping in DisEntitiesMap.Xml 149
Figure 67: BOM's Structure 151
Figure 68: LOCs and COCs of F-16 flight 153
Figure 69: LOCs and COCs of SA-8 SAM 153
Figure 70: External Interface of AddSIM 158
Figure 71: Simulation Connection Configuration of VR-Forces 159
Figure 72: SIMbox HLA extension 160
Figure 73: Loading SIMbox's LasVegas terrain format into VR-Forces 163
Figure 74: Screen Shot of SIMDIS 166
Figure 75: AddSIM Terrain 167
Figure 76: VR-Forces terrain database 168
Figure 77: SIMbox terrain database 169
Figure 78: Damage Value in VR-Forces 171
Figure 79: Situation Map and F-16 Flight, Mig-29 Flight, T-72 Tank and SA-8 SAM. 173
Figure 80: F-16C's attack to T-72 Tank Company 174
Figure 81: Damage Value of T-72 Tanks 175
Figure 82: The Damage Value according to the Distance of an Explosion 175
Figure 83: HLA Interface of OneSAF 186
Figure 84: DIS Interface of OneSAF 187
Figure 85: Terrain formats 195
Figure 86: Tree Diagram for the One-On-One Scenario (Single Shot) 201
Figure 87: VR-Exchange (Universal translator) Architecture 204

Figure 88: Recommended Actions from Case Study Findings and Lessons Learned 207

Figure 89: Common Standard-Defense Modeling and Simulation Process (CS-DMSP)211

Figure 90: Common Characters of Common Combat Entity 216

Figure 91: Common Operations of Common Combat Unit... 217

Figure 92: CS-SSAF 220

Figure 93: OneSAF AAR Architecture.. 225

Figure 94: CS-LSSAF... 235

Figure 95: CS-VSSAF 241

Figure 96: CS-CSSAF... 246

Figure 97: MIL-STD-252B Icons 247

Figure 98: Integration Process between RCS-ERC and RVTTS................................... 249

Figure 99: Common Standard-Correlated Terrain Database (CS-CTDB)...................... 251

Figure 100: Master Terrain Database Generation (TDB) Centers................................. 252

Figure 101: WebLVC Server 258

Figure 102: Short Term Strategy for LVC-ITA.. 259

Figure 103: Long Term Strategy for LVC-ITA 260

Figure 104: Live Virtual Constructive-Integrating Training Architecture (LVC-ITA).. 266

LIST OF TABLES

Table 1: Live, Virtual and Constructive Simulation Systems 8

Table 2: Purpose, Goal and Objectives of the Research 19

Table 3: Implications of LCIM 32

Table 4: Comparison of SSAs... 49

Table 5: Steps of FEDEP, SEDEP and DSEEP 53

Table 6: Research Questions.. 119

Table 7: Virtual Flight Simulator.. 137

Table 8: Virtual SAM Simulator... 137

Table 9: Constructive Simulation and Data Logger 138

Table 10: AddSIM 138

Table 11: LVC Simulation Test Criteria... 147

Table 12: Problems from LVC simulation case study results.. 148

Table 13: Three Object Component Types in SIMbox.. 152

Table 14: Coordinate systems... 165

Table 15: DIS Damage Appearance 172

Table 16: List of Lessons Learned.. 176

Table 17: Classification of Constructive Simulation System. 179

Table 18: Comparison of FEDEP, SEDEP and DSEEP 183

Table 19: CGF Comparison between OneSAF, VR-Forces and STAGE 184

Table 20: Raster formats... 192

Table 21: Vector Data Formats... 193

Table 22: Grid Data Formats 194

Table 23: List of the Recommended Actions 206

Table 24: Overview of Agile Roadmap for LVC-ITA 209

Table 25: Comparison about Main Technology in M&S Domain 255

LIST OF ACRONYMS (or) ABBREVIATIONS

AAC	Alarms and Alerts Component
A&I	Architecture and Integration
AAR	After Action Review
ACR	Advanced Concepts and Requirements
ACS	Aircraft Combat Survivability
ACS	Aural Cueing System
ACTF	Army Constructive Training Federation
ADD	Agency for Defense Development
AddSIM	Adaptive distributed parallel Simulation environment for Interoperable and reusable Models
ADS	Advanced Distributed Simulation
ALSP	Aggregate Level Simulation Protocol
ALT	Altitude
ANDEM	Architecture Neutral Data Exchange Mode
API	Application Programmers Interface
ARPA	Advanced Research Projects Agency
ASP	Aspect
ATIA-M	Army Training Information Architecture –Migrated
AVCATT	Aviation Combined Arms Tactical Trainer
BAX	Battle Area Complex
BCS	Battle Command Systems

BDA	Battle Damage Assessment
BOM	Base Object Model
BTB	Breathing Time Bucket
BTW	Breathing Time Warp
CANG	California National Guard
CATT	Combined Arms Tactical Training System
CBS	Corps Battle Simulation
CC	Constructive-Constructive
C2	Command and Control
C4I	Command, Control, Communications, Computers and Intelligence
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CCTT	Close Combat Tactical Trainer
CGF	Computer Generated Forces
CISE	Component based Integrated Simulation Engine
CM	Conceptual Model
CMMS	Conceptual Models of the Mission Space
COC	Console Object Component
COE	Contemporary Operational Environment
CONOPS	Concept of Operations
COP	Common Operating Picture
CORBA	Common Object Request Broker Architecture

COTS	Commercial-Off-The-Shelf
CPU	Central Processing Unit
CPX	Command Post Exercise
CS-CSSAF	Common Standard-Constructive Simulation System Architecture Framework
CS-CTDB	Common Standard-Correlated Terrain Database
CS-DMSP	Common Standard- Defense Modeling and Simulation Process
CS-LSSAF	Common Standard-Live Simulation System Architecture Framework
CSPAR	Common Standards, Products, Architectures and/or Repositories
C-SSA	Common Standard Simulation Architecture
CS-SSAF	Common Standard-Simulation System Architecture Framework
CS-VC	Common Standard-Virtual Components
CS-VSSAF	Common Standard-Virtual Simulation System Architecture Framework
CTC	Combat Training Center
CTIA	(US Army's) Common Training and Instrumentation Architecture
CTEIP	Central Test and Evaluation Investment Program
CVEM	Common Virtual Environment Management
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management System
DCMF	Defense Conceptual Modeling Framework
DCP	Data Collection Plan
DDM	Data Distribution Manager
DEM	Digital Elevation Model

DFIRST	Deployable Force-on-Force Instrumented Range System
DIS	Distributed Interactive Simulation
DIST	Distance
DLC	Dynamic Link Compatibility
DLL	Dynamic Link Library
DMA	Defense Mapping Agency
DM&S	Defense Modeling and Simulation
DMPRC	Digital Multi-Purpose Range Complexes
DMPTR	Digital Multipurpose Purpose Training Range
DMSO	Defense Modeling and Simulation Office
DoD	U.S. Department of Defense
DSEEP	Distributed Simulation Engineering and Execution Process
DTED	Digital Terrain Elevation Data
DVD	Digital Video Disc
DVED	Database Virtual Environment Development
EDM	Environmental Data Models
ERC	Environmental Runtime Component
ERF	Entity Resolution Federation
FCS	Future Combat System
FDB	Feature Database
FEDEP	Federation Development and Execution Process
FLIR	Forward Looking Infrared Radar

FOI	Swedish Defense Research Agency
FOM	(HLA) Federation Object Model
FTI	Fixed Tactical Internet
EXCON	Exercise Control
GEF	Graphical Editing Framework
GEH	Global Event Horizon
GIG	Global Information Grid
GIS	Geospatial information systems
GOTS	Government-Off-The-Shelf
GPS	Global Positioning System
GRIB	GRIdded Binary
GUI	Graphical User Interface
HDD	Hard Disk Drive
HDG	Heading
HITS	Homestation Instrumented Training Systems
HLA	High Level Architecture
ICD	Initial Capability Document
IEEE	Institute of Electrical and Electronics Engineers
IEMS	Industrial Engineering and Management Systems
IG	Image Generator
I-MTS	Integrated - Military Operations in Urban Terrain (MOUT) Training Systems
IOS	Instructor Operator Station

ISC	Instrumentation Control
ISO	International Organization for Standardization
JCATS	Joint Combat and Tactical Simulation
JIIM	Joint, Interagency, Intergovernmental and Multinational
JLCCTC	Joint Land Component Constructive Training Capability
JLVCDT	Joint Live Virtual Constructive Data Translator
JLVC-TE	Joint Live Virtual Constructive-Training Environment
JNTC	Joint National Training Capability
JSAF	Joint Semi-Automated Forces
JSON	JavaScript Object Notation
JTEN	Joint Training and Experimentation Network
JTEP	Joint Training Experimentation Program
KA	Knowledge Acquisition
KM	Knowledge Modeling
KR	Knowledge Representation
KU	Knowledge Use
LAN	Local Area Networks
LCD	Liquid Crystal Display
LCIM	Levels of Conceptual Interoperability Model
LEH	Local Event Horizon
LOC	Logic Object Component
LROM	Logical Range Object Model

LRU	Line Replaceable Unit
LT2	Training Transformation
LT2-FTS	Live Training Transformation - Family of Training Systems
LVC	Live Virtual Constructive
LVCAR	Live Virtual Constructive Architecture Roadmap
LVC-IA	Live, Virtual, Constructive-Integrating Architecture
LVC-ITA	Live Virtual Constructive-Integrating Training Architecture
LVCWA	Live Virtual Constructive Architecture Way Ahead
MC	Mission Command
M&S	Modeling & Simulation
MDB	Master Database
MGRS	Military Grid Reference System
MIL-STD	The U.S. Military Standard
MOE	Measures of Effectiveness
MOP	Measure of Performance
MOUT	Military Operations in Urban Terrain
MOUT	Mobile Operations on Urban Terrain
MPI	Multi Passing Interface
MRF	Multi-Resolution Federation
MRM	Multi Resolution Method
MRX	Mission Rehearsal Exercise
MSM	Mission Space Models

MTC	Mission Training Center
M-TDB	Master Terrain Database Generation
NATO	North Atlantic Treaty Organization
NECC	Net Enabled Command Capability
NetCDF	Network Common Data File
NGA	National Geospatial-Intelligence Agency
ODD	Optical Disc Drive
OIS	Objective Instrumentation Systems
OMT	(HLA) Object Model Template
OneSAF	One Semi-Automated Forces
OOS	OneSAF Objective System
ORD	Operational Requirements Documents
OS	Operating System
OSAMS	Open Simulation Architecture for Modeling and Simulation
OSD	Office of the Secretary of Defense
OSG	OpenSceneGraph
PEO-STRI	(U.S. Army) Program Executive Office for Simulation, Training and Instrumentation
PDU	(DIS) Protocol Data Unit
PDMS-SSG	Parallel and Distributed Modeling & Simulation Standing Study Group
PDT	Participant Definition Tool
PLA	Product Line Architecture

PLAF	Product Line Architecture Framework
PLAS	Product Line Architecture Specification
PLM	Product Line Management
PM TRADE	Program Manager for Training Devices
PoE	Panel of Experts
PVD	Plan View Display
R&D	Research & Development
RDA	Research, Development, and Acquisition
RDGT	Real-Time Database Generation Toolkit
ROK	Republic of Korea
ROK AF	Republic of Korea Armed Force
RPR FOM	Real-Time Platform Reference Federation Object Model
RCS	Representative Constructive Simulation
RTI	(HLA) Run Time Infrastructure
RVTTS	Representative Virtual Tactical Training Simulator
SAF	Semi-Automated Forces
SAM	Surface to Air Missile
SC	Steering Committee
SDA	(TENA) Software Development Activity
SDK	Software Development Kit
SEDEP	Synthetic Environment Development and Exploitation Process
SE Core	Synthetic Environment Core

SEDRIS	Synthetic Environment Data Representation and Interchange Specification
SIMNET	SIMulation NETworking
SISO	Simulation Interoperability Standards Organization
SIL	Simulation Interoperability Laboratory
SISO	Simulation Interoperability Standards Organization
SOA	Set of Actions
SOM	(HLA) Simulation Object Model
SPD	Speed
SSA	Standard Simulation Architecture
STANAG	Standardization Agreement
STDGC	Standard/Rapid Terrain Generation Capability
STF	SEDRIS Transmittal Format
TACSIM	Tactical Simulation
TADSS	Training Aids, Devices, Simulations, and Simulators
TAF	Tactical Analysis and Feedback
TDB	Terrain Database
TE	Testing and Evaluating
TEMO	Training, Exercises, and Military Operations
TENA	Test and Training Enabling Architecture
TESS	Tactical Engagement Simulation Systems
TIFF	Tagged Image File Format
TNS	Tactical Net Selector

TRADE	Training Devices
TSP	Training Support Packages
UCF	University of Central Florida
UDC	User Defined Code
U.S.	United States
UTM	Universal Transverse Mercator
VBS2	Virtual Battlespace 2
VC	Virtual and Constructive
VGA	Video Graphics Array
VSA	Virtual Simulation Architecture
VV	Virtual-Virtual
WAN	Wide Area Network
WARSIM	Warfighters' Simulation
WFX	Warfighter Exercise
WWW	World Wide Web
XML	EXtensible Markup Language

CHAPTER ONE: INTRODUCTION

Live, Virtual and/or Constructive simulation systems (or federates) have emerged as a flexible and cost-effective solution for training, acquisition and analysis. Live, Virtual and/or Constructive simulations are of importance in the military domain as well as in industries. Today's advanced Modeling and Simulation (M&S) technologies have been developed towards the goal of seamless interaction between the Live, Virtual and/or Constructive simulation systems. Usually, "Live Virtual Constructive (LVC)" refers to the combination of three types of distributed simulation systems and applications into a single distributed system. Although today's M&S technologies such as the high speed networking and Simulation Standards Architectures (SSA) (or Simulation Interoperability Protocol) allow trainees to participate in LVC simulation environments restrictively, there are lots of things that still must be addressed. In the results, the many advantages of LVC training are currently limited by lack of full interoperability with other Live, Virtual and/or Constructive simulation systems and Battle Command Systems (BCS).

Currently, a number of SSAs are commonly used. The typical Live, Virtual, and/or Constructive SSAs in-place today are Aggregate Level Simulation Protocol (ALSP), Distributed Interactive Simulation (DIS), High Level Architecture (HLA), Test and Training Enabling Architecture (TENA), and Common Training Instrumentation Architecture (CTIA). Each of the SSAs was developed by particular M&S user communities to meet their specific needs or requirements. Although each of the Live, Virtual and/or Constructive simulation systems (or federates) rely on a specific SSA to exchange data in distributed simulation environments;

regrettably, Live, Virtual, and/or Constructive simulation systems that choose different SSAs cannot be natively interoperable with each other (Henninger et al., 2008).

Eventually, this serious issue is directly linked to LVC interoperability and integration. There has been prior research to solve these problems and to improve interoperability between different SSAs. However, most research has been focused on developing new LVC SSA capable of interoperability regardless of LVC simulation systems through different SSA. Developing new LVC SSA and single SSA convergence would be the long-term strategy. Although a new LVC SSA would attain the goal of LVC interoperability and integration, to some degree, it cannot be prepared for all future problems. Therefore, we concluded that migrating to single LVC SSA was impractical in the near future, and multi-SSAs simulation environments would remain the state of the practice for the foreseeable future.

For these reasons, I have considered a need for an agile roadmap which reflects user's situational needs and expectations to decrease the complexity of the integration and increase the interoperability of the LVC simulation systems.

This study is to suggest an agile roadmap for the Live Virtual Constructive – Integrating Training Architecture (LVC-ITA) pursuing the simpler integration, cost-effective, shorter user time and a flexible solution that will address these problems and introduce interoperable LVC simulations. I define that LVC-ITA is a set of common, standards Live, Virtual and Constructive simulation architecture framework that support a seamless and interoperable, integrated LVC environment where common hardware, software and network components and modules are interchangeable with other LVC components. The goal of the LVC-ITA is to seamlessly interconnect and ensure interoperability with other LVC simulation systems.

Chapter 1 briefly summarizes the general background, statement of the problems, purpose, goal and objectives of the research and expected contributions.

1.1 General Background

This section is to provide general information of (a) Stand-alone Simulation and Distributed Simulation, (b) Concept of Live, Virtual or Constructive Simulations, (c) Concept of Live Virtual Constructive (LVC) Simulations, (d) Overview of Standard Simulation Architecture (SSA) and then, (e) Overview of the U.S. Army Live, Virtual, Constructive-Integrating Architecture (LVC-IA).

1.1.1 Stand-alone Simulation and Distributed Simulation

In general, simulation systems can either be *stand-alone* as shown in Figure 1, or they can be used as a distributed system that runs different simulation systems at the same time as shown in Figure 2. Originally, many simulation systems were designed as stand-alone systems. Figure 1 depicts the fundamental concept that a stand-alone simulation is designed to simulate a complex model while operating independently, without interacting with other simulations.

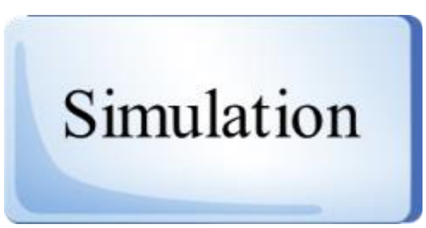


Figure 1: Stand-alone simulation.

Advanced Internet technology made possible the networking of computers located at geographically distributed sites. The development of supporting protocols and architectures has led to widespread use of a *distributed simulation*. The distributed simulation is concerned with the execution of simulations on geographically distributed computers interconnected via a Local Area Network (LAN) and/or Wide Area Network (WAN) (Fujimoto, 1999). Figure 2 illustrates the concept that a distributed simulation is designed to simulate a complex model by enabling distributed simulation components. Figure 2 also represents the larger complex system being modeled as a distributed simulation system. Generally speaking, characteristics of a typical distributed simulation include:

- It is geographically distributed.
- It may contain very large and complex software components.
- It may interact with concurrent Live (or real) systems.
- Its capability is subject to the constraints of computation resources (e.g. memory, Central Processing Unit (CPU), network speed), level of complexity and size.

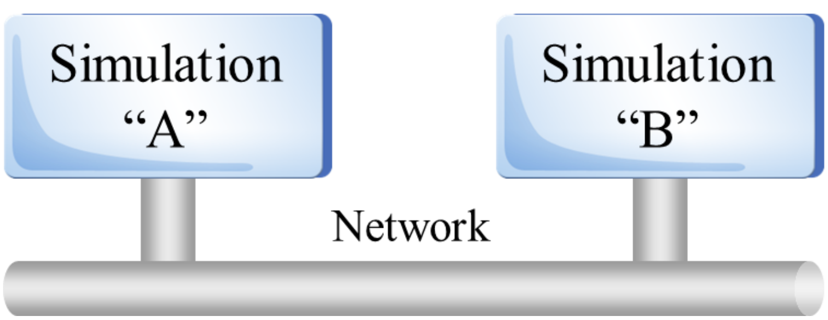


Figure 2: Distributed simulation.

The distributed simulation provides several advantages as compared to the stand-alone simulation systems. First, executing the simulation program on a set of geographically distributed computers enables one to create virtual worlds with multiple participants who are physically located at different sites (Fujimoto, 1999). In addition, it facilitates efficient use of past M&S assets developed by different manufacturers, as new, very powerful simulation environments can be quickly configured from existing M&S assets. Finally, it provides flexible mechanisms to integrate hardware and/or live assets into a unified environment for training or testing, and it is much more scalable than stand-alone systems (APL, 2010).

1.1.2 Concept of Live, Virtual or Constructive Simulations

Military simulation systems can be classified as belonging to one of three different types of simulation systems - Live, Virtual, or Constructive. A broadly used taxonomy for classifying simulation types (*MODELING AND SIMULATION (M&S) MASTER PLAN*, 1995).

1.1.2.1 Live simulation.

A simulation system involves real people operating real equipment or systems in a real environment not a virtual environment (e.g. a pilot flying a flight as shown in Figure 3).

Figure 3: Live simulation: U.S. Air Force F-22 Raptor

Source: http://www.af.mil/News/Photos.aspx?igphoto=2000930202

1.1.2.2 Virtual simulation.

A simulation system involves real people operating simulators / emulators / operational systems in a synthetic environment (e.g. a pilot flying a simulated flight as shown in Figure 4).

Figure 4: Virtual simulation: F-16 Mission Training Center (MTC)

Source: http://www.f-16.net/f-16-news-article4125.html

1.1.2.3 Constructive simulation.

A simulation system involves that simulated people operating simulated systems in a simulated environment (e.g. a simulated pilot flying a simulated flight as shown in Figure 5). Real people provide inputs (make inputs) to such simulation systems, but are not involved in determining the outcomes.

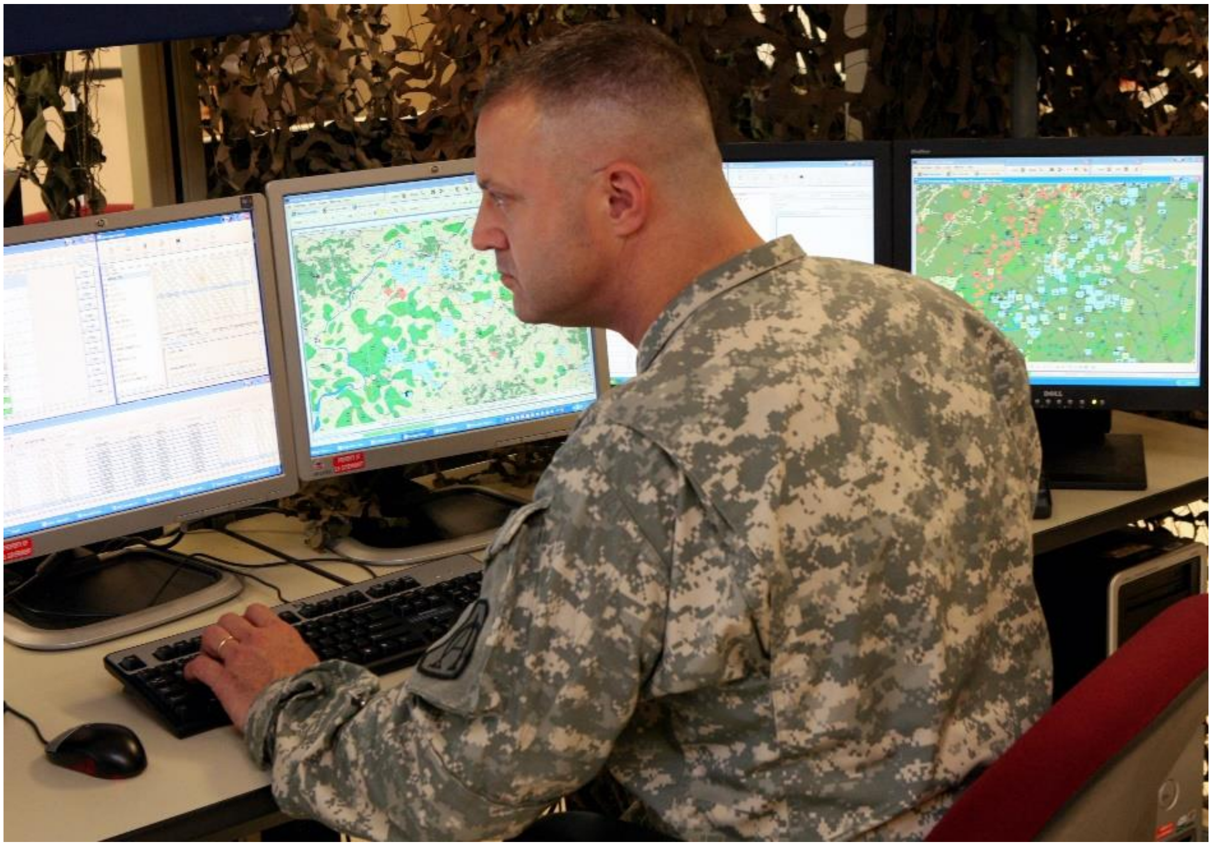


Figure 5: Constructive simulation

Source: http://asc.army.mil/web/portfolio-item/peo-stri-joint-land-component constructive-training-capability-jlcctc/

Table 1 summarizes the explanation of Live, Virtual, and Constructive simulation systems as was stated above.

Table 1: Live, Virtual and Constructive Simulation Systems

Category	Live	Virtual	Constructive
People	Real	Real	Simulated
Systems	Real	Simulated	Simulated

1.1.3 Concept of Live Virtual Constructive (LVC) Simulations

Virtual and Constructive simulations can be used in tandem with Live simulations in what is called Live Virtual Constructive (LVC) simulations. Usually, "LVC" refers to the combination of three types of distributed simulation systems and applications into a single distributed simulation system. The goal of LVC is to combine Live (or real), Virtual, and Constructive assets into one seamless and coherent environment operating in real time (Tolk, 2012). A graphical representation of an LVC synthetic environment is shown in Figure 6.

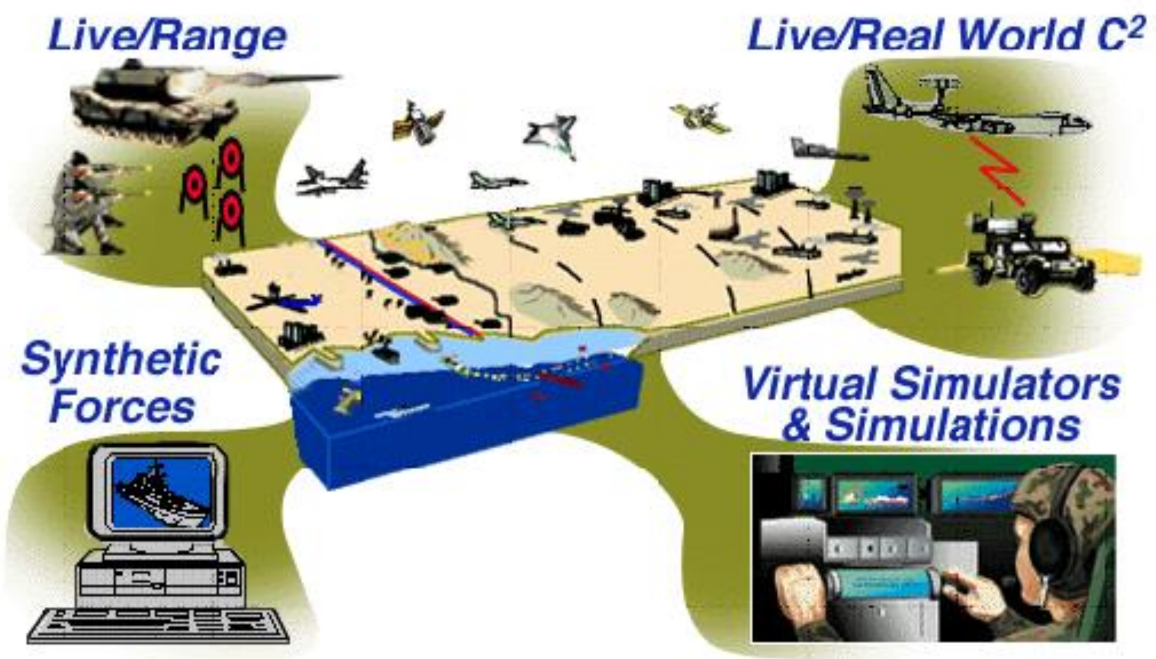


Figure 6: Graphic of an LVC Synthetic Environment

Source: W. Bizub, Bryan, and Harvey (2006)

If that is so, why do several simulation systems need to be connected? The systems of an organization are needed to work in conjunction with the systems of other organizations. M&S communities may already have a number of simulation systems that need to be used together with newly acquired simulation systems or simulation systems from other organizations. Another reason is that there may be a requirement to simulate a “bigger picture,” where models from different organizations interact. Experts from different fields need to contribute different models. In many cases, it would also be a monumental task to build one big system that covers everything compared to connecting several different simulation systems (Mller, 2013).

If the goal of LVC is achieved, M&S users can get the benefit as mentioned below.

First, from the cost aspect, LVC simulations can now be conducted at a lower cost as they limit the unnecessary movement of troops and equipment (Tolk, 2012). Pure simulation systems are inherently less expensive than a live event with real assets. Without a doubt, cost saving is one of the primary reasons to simulate real systems, instead of simply using the real systems themselves (Noseworthy, 2008).

Second, from the effectiveness aspect, LVC Simulations can provide cost-effective, repeatable, and quantitatively analyzable means of “practicing” different scenarios. Scenarios range from tactical levels to joint/coalition strategic levels involving members from every branch and rank in the military hierarchy (Andreas, Saikou, & Charles, 2007).

Finally, from the training tool aspect, LVC training simulation systems can provide warfighters the ability to train as a team, while supporting the enhancement of individual proficiency. The primary focus is comprehensive tactical training for all warfighters. The main goal of LVC simulation is always to train them on mission essential competencies needed for combat readiness.

In conclusion, much research for a seamless LVC simulation that is the LVC-IA, Live Virtual Constructive Architecture Roadmap (LVCAR), Synthetic Environment Core (SE Core) programs, and Future Combat System (FCS), will gradually eliminate many of the shortfalls, leading to a training environment that more closely replicates the operational environment (Shufelt Jr, 2006).

1.1.4 Overview of Standard Simulation Architecture (SSA)

A number of Standard Simulation Architectures (SSAs) are commonly used today. SSAs have been developed in order to achieve interoperability among independently developed simulation systems. SSAs are intended to allow independently executing models to interoperate, via a network, so as to collaboratively simulate a common scenario or environment. Each of the SSAs can include definitions of the formats of the messages to be exchanged at runtime between the linked models, the data items contained in those messages, and the logical actions and sequences to be performed when models interact via those messages (Tolk, 2012).

Currently, SSA is called different names: (a) *Distributed Simulation Architecture* (Fujimoto, 1999; Henninger et al., 2008; Loper & Cutts, 2010), (b) *Modeling and Simulation (M&S) Interoperability Standards* (Tolk, 2012), (c) *M&S Interoperability Protocol* (Granowetter, 2013), (d) *Distributed Simulation Protocol* (Zalcman, Blacklock, Foster, & Lawrie, 2011) or (e) *Simulation Architecture* (Gustavsson, Björkman, & Wemmergård, 2009), etc. In this research, according to Jeffrey S Steinman and Hardy (2004), I standardize the terminology of the above different names as ***Standard Simulation Architecture (SSA)*** in order to avoid confusion in the rest of the thesis.

Today, a number of SSAs have been used but the main SSAs developed in the U.S. that were considered in this research include: ALSP, DIS, HLA, TENA and CTIA. The presence of multiple SSAs allows users to select the SSA that best meets their needs (O'Connor et al., 2006). These SSAs that evolved by specific user communities have matured based on changing user requirements. These SSAs have all contributed to a distributed simulation environment where highly-distributed training, mission rehearsal, operations support, and joint/coalition exercises have become a reality (Mittal, Doyle, & Portrey).

1.1.4.1 Historical Evolution of the Standard Simulation Architectures (SSAs).

Figure 7 illustrates the historical relationships and content of the Standard Simulation Architectures (SSAs). Arrows in the figure, indicate ideas and experience flowing from one SSA to the benefit of the next, but they do not necessarily mean that one SSA is replacing or subsuming another. In fact, five (ALSP, DIS, HLA, CTIA and TENA) of the six SSAs remain in active use as shown in Figure 7 (Tolk, 2012).

Defense Advanced Research Projects Agency (DARPA) sponsored the Simulation Networking (SIMNET) program which started in 1983, and ALSP program started in 1989. The SIMNET is no longer used, but the SIMNET evolved and matured into the DIS. In the mid-1990s, the Defense Modeling and Simulation Office (DMSO) started the HLA program to combine the best features of DIS and ALSP into a single SSA that could also support uses in the analysis and acquisition as well as training applications. Particularly, in the 2000s, two communities started development of alternate SSA due to HLA's unacceptable performance limitations. The real-time test range community started development of the TENA to integrate Live assets in the test-range setting. Similarly, the U.S. Army started development of CTIA to interconnect Live assets on Army training ranges. All of these architectures except SIMNET remain in service and is still evolving.

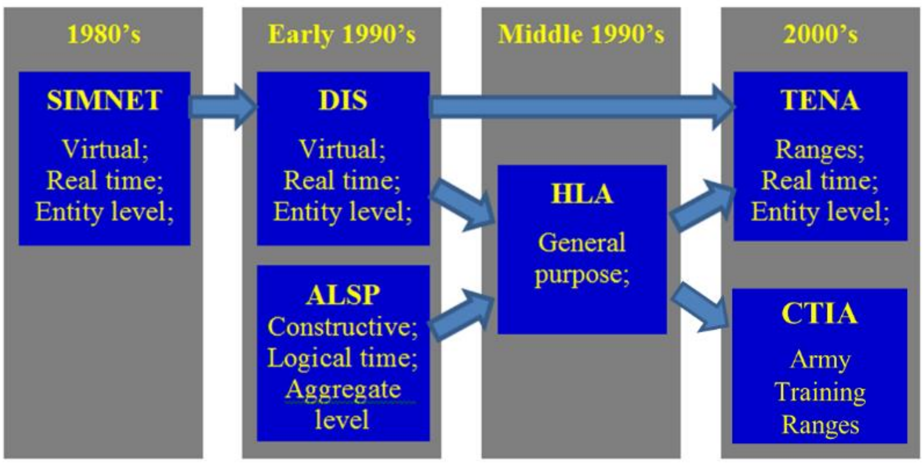


Figure 7: Historical Evolution of Standard Simulation Architecture (SSA)

1.1.5 Overview of the U.S. Army Live, Virtual, Constructive-Integrating Architecture (LVC-IA)

The U.S. Army LVC-IA is a critical component of the Army's training transformation. It is a network-centric linkage that collects and assimilates information between Live and simulation instrumentation (Haight, 2007). The U.S. Army LVC-IA will provide the foundational structure and framework for integrating LVC systems into the Integrated Warfighter's Training Environment as shown in Figure 8. The objective of LVC-IA is to enable on-demand training, mission planning and rehearsals, C4ISR interaction, and Joint, Interagency, Intergovernmental and Multinational (JIIM) interoperability anytime and anywhere. The U.S. Army LVC-IA is a set of protocols, specifications, standards, and services/infrastructure that support the operation of a seamless and integrated LVC environment where hardware, software, network components, and modules are interoperable with other LVC components and the BCS (Black, Brown, Levine, & Sudnikovich, 2008). More detailed information is described in Chapter 2.

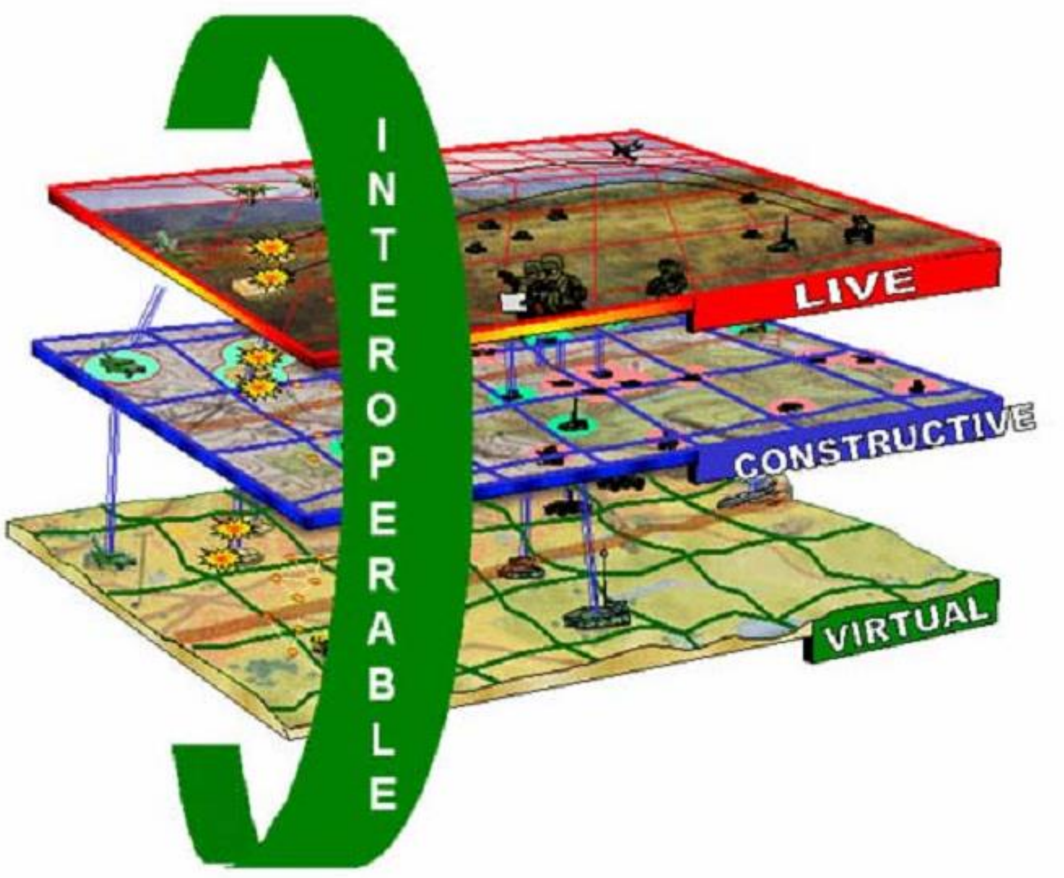


Figure 8: LVC-IA

Source: Black et al. (2008)

1.2 Statement of the Problems

In history, most distributed simulations have been more or less homogeneous. Therefore, people have typically put together exercises where everyone used DIS, or where everyone used HLA. In the last couple of years, however, things have changed: distributed simulations are more often being put together from existing assets, which have been built, tested, and verified against some set of pre-existing SSA choices. In addition, large exercises are becoming more common, with multiple sites connected over a WAN. These exercises are widely distributed not only in the sense of geography and network topology. Each site manager might want to make his own decisions about what SSAs to use. They still want to be able to integrate with other sites easily on short time and at a small outlay (MÄ K).

While there is more integration between Live, Virtual, and/or Constructive assets, it is well recognized in the M&S community that there is limited interoperability between them, loose integrated Live, Virtual and/or Constructive assets through multiple SSAs, multiple type existences of SSAs and complex technical tools for LVC simulation. In spite of much improvement in M&S interoperability since the advent of SSAs in the 1980s, there is ~~a~~ limited interaction between the Live, Virtual and/or Constructive and many problems exist with respect to the procedures and technologies to improve interoperability and integration between Live, Virtual and/or Constructive assets. That means, up till now, most participants in distributed simulation exercises would normally only be expected to be Virtual, Constructive or Virtual and Constructive (VC) systems.

1.2.1 Problem 1: Inherent Limited Interoperability between the Different SSAs.

The SSAs in place today are ALSP, DIS, HLA, TENA, and CTIA (Fujimoto, 1999; Loper & Cutts, 2008). In other words, only one universally agreed-upon SSA is not yet available. Originally, Live, Virtual and/or Constructive simulation systems were NOT developed to interoperate with each other. Although SSAs are developed to make simulation systems ~~to~~ interact with each other across network connections, ALSP, DIS, HLA, TENA, and CTIA simulation systems (or federates) are not inherently interoperable with each other (Loper & Cutts, 2008). Naturally, Live, Virtual and/or Constructive simulation systems (or federates) that choose different SSAs inherently cannot interoperate. When more than one SSA must be used in the same simulation environment (or federation), interoperability problems are compounded by the architectural differences. For example, HLA and DIS are most often used for integrating Virtual and Constructive (VC) assets but HLA or DIS are not particularly well suited for real-time Live systems (Noseworthy, 2008). Meanwhile, TENA is widely used in testing and to integrate Live assets into exercises/events. Marsden, Aldinger, and Leppard (2009) researched the interoperability between TENA and DIS training architectures. Although TENA and HLA are similar in some aspects, their native incompatibility is a major inhibitor to seamless LVC interoperability (W. W. Bizub & Cutts, 2007). CTIA promotes commonality among the U.S. Army's instrumented ranges and home stations.

Even if simulation systems were combined as a collection of enterprise within an HLA federation, communication between such simulation systems is often sporadic and irregular. Thus, the incompatibilities between these SSAs require spending a considerable amount of resources and time to develop point solutions that efficiently integrate them into a single, unified set of supporting simulation services. One benefit of having only one common SSA is that simulation systems and services make use the same programming constructs and can be therefore, more freely interoperable (Jeffrey S. Steinman, 2013).

1.2.2 Problem 2: Many Issues in Integrating LVC Assets.

Furthermore, when simulations are connected between different SSAs, additional steps must be taken to ensure effective information is exchanged between all applications. In most cases, these additional steps typically involve interposing bridges, gateway application and data exchange models between the multiple SSAs for the limited level of LVC interoperability (Loper & Cutts, 2008).

However, these solutions to technical interoperability may result in significantly violating latency thresholds, increased risk, complexity, cost, data mistranslation, disconnect, level of effort and inflexibility and preparation time with multiple SSAs (Loper & Cutts, 2008). The increased complexity of distributed simulation systems tends to increase the likelihood that a software defect will cause at least some part of the system to malfunction (Tolk, 2012). In addition, the cost of failure may be little more than the inconvenience of restarting the simulation systems. Certainly, the lost time may be significant, and this in turn may result in a cost of corresponding significance (Noseworthy, 2008). The worst situation is that the data mistranslation coming from using different SSAs may produce erroneous simulation results without notice.

Thus, the inherent limited interoperability between the different SSAs introduces a significant and unnecessary barrier to the integration of Live, Virtual, and/or Constructive simulation systems. This barrier must be significantly reduced or eliminated. To solve these problems, M&S user communities require the development of a new LVC SSA or point solutions that should be highly interoperable regardless of whether the simulation systems are Live, Virtual or Constructive.

1.2.3 Problem 3: Decentralized Development of SSAs and LVC Assets

These are fundamental environment characteristics of large-scale LVC simulation environments. As the number of mixed-SSA events increases over time, the inter-SSA communication problem increases as well. In addition, the development of many simulation applications and SSAs is also decentralized. In general, distributed simulations are typically made up of a variety of simulation applications. In particular, each Live, Virtual and/or Constructive simulation system consists of a single application homogeneously used throughout all the systems. This is a result of the wide differences in the nature of the applications that support Live systems, the applications that create Virtual simulators, and the applications that create Constructive simulation systems. The more the LVC simulation system is large-scale, the more the number of distinct applications used throughout the entire LVC simulation system increases. As a result, the development is nearly always decentralized, lacking a common authority to provide a uniform and consistent development process (Noseworthy, 2008).

1.3 Purpose, Goal and Objectives of the Research

Table 2 summarizes the purpose, goal and objectives of this research. There are four objectives to achieve the goal. Four objectives that are specific steps that were taken to meet the goal are described in detail.

Table 2: Purpose, Goal and Objectives of the Research

Purpose	To enhance the interoperability, integration, composability and reuse in LVC simulation environment.
Goal	Providing an agile roadmap for the LVC-ITA
Objectives	1. Assessment of the current state in an LVC simulation environment.
	2. Making the right vision for the future M&S
	3. Conducting a case study reflecting current LVC simulation situation.
	4. Drawing lessons learned from the case study.

The ultimate purpose of this research is to enhance the interoperability, integration, composability and reuse in LVC simulation environment. To achieve the purpose, the goal of this research is to provide an *agile roadmap for the Live, Virtual, Constructive-Integrating Training Architecture (LVC-ITA)*. LVC-ITA is a set of common and standards Live, Virtual and Constructive simulation architecture framework that supports a seamless and interoperable, integrated LVC environment where common hardware, software and network components and modules are interchangeable with other LVC components.

There is much good research written on the topics of LVC simulation and distributed simulation, and most of them are used as references in this research. So why is this additional research on LVC-ITA topic necessary? The reason is simple: while all other research in this domain successfully highlighted special topics in detail, none of them compiles the knowledge of all contributing fields that newly M&S committee needs to consider, in particular, to achieve LVC-ITA. Therefore, the objective of this research is to provide an *Agile Roadmap for the LVC-ITA* to guide the newly M&S communities to find solutions that will address the problems mentioned above as well as tasks for LVC-ITA, and that it results in the increase of the level of interoperability, integration, composability and reuse.

1.3.1 Objective 1: Assessment of the Current State in an LVC Simulation Environment.

This research is to investigate the issues related to LVC interoperability, integration, composability and reuse for future LVC simulation thoroughly.

● First, I assess the current state, including existing tools, technologies, methodologies, existing interface, etc.

● Second, I compare and contrast the development, evolution processes and types for the five SSAs (ALSP, DIS, HLA, TENA and CTIA) used by each M&S community.

● Third, I identify previous the U.S. DoD LVCAR's recommended approach, the U.S. Army LVC-IA and other related works.

● Last, I draw the rationale for improvements based on their research.

1.3.2 Objective 2: Making the Right Vision for the Future M&S

From Goal 1, I can establish a roadmap for accomplishing the purpose of this research after making the right vision for the future M&S. Traditional approaches based on ad-hoc development from several organizations and software program cannot accomplish our research purpose. The vision includes cost-effectiveness, easier to use and maintain, feasible technology, network-centric, high quality and reliability, multiple-use concepts and composability, etc.

1.3.3 Objective 3: Conducting a case study reflecting current LVC simulation situation

This research illustrates the case study using an Adaptive distributed parallel Simulation environment for Interoperable and reusable Model (AddSIM) that is component based integrated simulation engine that was developed by the Agency for Defense Development (ADD) in South Korea. Through the case study, I can draw lessons learned to apply to an agile roadmap for the LVC-ITA.

- First, I analyze and evaluate the current capabilities of AddSIM.
- Second, I study a technical approach for how the AddSIM, and the newest technologies can be applied to the case study.
- Third, I plan to integrate Virtual and Constructive assets into the AddSIM, and then, a conceptual Live asset is ported into AddSIM.
- Last, LVC simulation case study is executed, then I find some problems from the case study.

1.3.4 Objective 4: Drawing lessons learned from the case study.

From objectives 2 and 3, the last objective is to draw lesson learned that can be applied to the roadmap from the case study across the multiple SSAs and Live, Virtual and Constructive assets.

● First, I seek the desired future LVC-ITA for M&S user-centered requirement.

● Second, I analyze and evaluate the case study results.

● Last, I draw lessons learned and I lay the foundation for developing an efficient roadmap to maximize technical interoperability, integration, composability and reuse of LVC simulation across M&S communities.

1.4 Expected Contributions

This research provides a roadmap to enable the newly emerging M&S communities to begin to make progress towards highly interoperable LVC simulation environments. The roadmap tries to provide adequate discussion of the major issues and applicable solutions associated with each issue. Therefore, my dissertation will guide the discussion of each issue to an implementable, technically feasible, and affordable solution considering several options and making an appropriate choice.

The results, if the M&S developing communities adopt the agile roadmap for LVC-ITA, benefits of such a roadmap would be: (a) support for scalable distributed simulation due to simpler and flexible integration, (b) significantly improved interoperability, composability and reuse between simulation systems, and (c) considerably cost-effective and rapid manner due to simple integration for M&S community.

1.4.1 Contribution 1: Simpler Integration

The first research contribution is the simpler integration. Advanced techniques, tools, and simulation architecture frameworks are needed to reduce the complexity of developing technologies in the emerging LVC simulation. Thus, to integrate an additional application without requiring changes to the existing native federates is possible. The other integrated applications can continue as before.

1.4.2 Contribution 2: Flexible Integration

Another research contribution is the flexible integration. An agile roadmap for the LVC-ITA will be a more flexible integration approach than the traditional solution to integration. The existing SSAs are so easily integrated that they can be viewed as a single SSA. This is enough to support interoperability regardless of the SSA being used in the target simulation systems (e.g., DIS, HLA 1.3, HLA 1516, HLA Evolved, TENA and CTIA, etc.), without requiring changes to the existing native simulation systems. The agile roadmap for the LVC-ITA makes it easier for M&S users to adapt to the new protocols.

1.4.3 Contribution 3: Reuse Legacy Simulation Systems

In terms of reuse, the contribution of my work is also in facilitating reuse legacy simulation systems. According to a previous LVCAR study (Henninger et al., 2008), one of the main fundamental guidelines is "Do No Harm" which means that the DoD should NOT take any immediate action to discontinue any of the existing SSAs. Therefore, the agile roadmap of the LVC-ITA is likely to use an existing SSAs such as DIS, HLA, TENA or CTIA.

1.4.4 Contribution 4: Cost-effective and Shorter Time to LVC User

Finally, there is one other contribution that should be of an impact on M&S community. LVC users need short term solutions that reduce both cost and technical complexity and risk until such time as SSA convergence can be achieved. In other words, the roadmap for LVC-ITA must provide a strategy for achieving the purpose in a rapid, efficient and flexible manner. To satisfy new requirements from LVC user communities, applications can be integrated in new ways. As mentioned of above, this will take less time for users because integration is simpler. The roadmap includes recommended actions such as improved bridging tool, common simulation architecture framework, and common object models. An agile roadmap can provide significant near and midterm value to the M&S user community.

1.5 Dissertation Outline

The dissertation is organized as six chapters in total. The rest of chapters are organized as follows:

● Chapter 2 Literature Review.

● Chapter 3 Methodology.

● Chapter 4 Case Study.

● Chapter 5 Agile Roadmap for LVC-ITA.

● Chapter 6 Conclusion.

Chapter 2 contains detailed information on (a) efforts for improving LVC interoperability, (b) interoperability, integration and composability, (c) comparison of SSAs, (d) conceptual model (CM), (e) bridging solutions, (f) U.S. DoD LVCAR, (g) U.S. Army LVC-IA, and (h) CSPAR.

Chapter 3 describes the methodology for the development of an agile LVC-ITA roadmap. It discusses the step-by-step processes.

Chapter 4 describes an LVC simulation case study using the AddSIM. First, I examined the newest technologies to apply to the case study reflecting current LVC simulation environments. Second, I planed the case study. Third, I described the findings from the case study. Fourth, based on the findings, lessons learned as well as discussions are provided. Last, the recommended actions to meet the lessons learned from the case study are described.

Chapter 5 proposes an agile LVC-ITA roadmap developed from recommended actions.

Chapter 6 provides an overall discussion of the dissertation, summarizes the conducted research, highlights the contributions and discuss limitations for future work.

Figure 9 shows the overview of the dissertation.

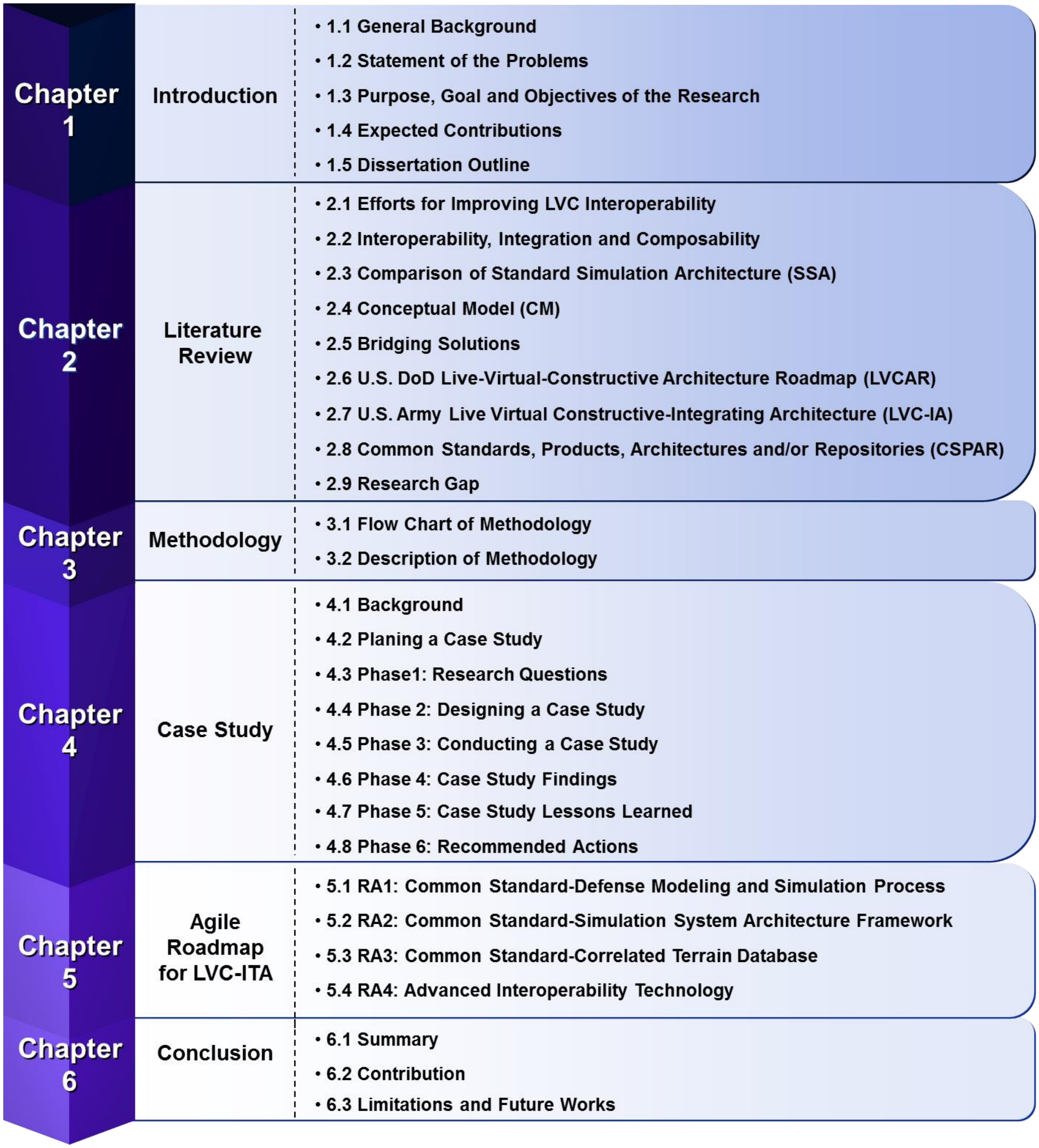


Figure 9: Overview of the Dissertation

CHAPTER TWO: LITERATURE REVIEW

By necessity, the agile roadmap covers a number of related topics that should work together for LVC-ITA. Therefore, this chapter summarizes the research on (a) efforts for improving LVC interoperability, (b) interoperability, integration and composability, (c) comparison of SSAs, (d) conceptual modeling, (e) bridging solutions, (f) U.S. DoD LVCAR, (g) U.S. Army LVC-IA, and (h) Common Standards, Products, Architectures and/or Repositories (CSPAR). Research gaps are then identified.

2.1 Efforts for Improving LVC Interoperability

There has been much research for improving LVC interoperability. One possible approach includes adopting a single, agreed-upon architecture for the simulation environment. Another approach is developing a point solution between the multiple SSAs. Currently, technical interoperability has been achieved through a number of methods, including the use of gateways and bridges, etc.

2.1.1 Department of Defense (DoD) Modeling and Simulation (M&S) Master Plan

In 1995, Department of Defense (DoD) represented *Modeling and Simulation (M&S) Master Plan* to address the full range of issues confronting DoD M&S. This plan shows the six objectives and the breakout of the objectives into sub-objectives to facilitate interoperability and reuse as shown in Figure 10 (DoD, 1995).

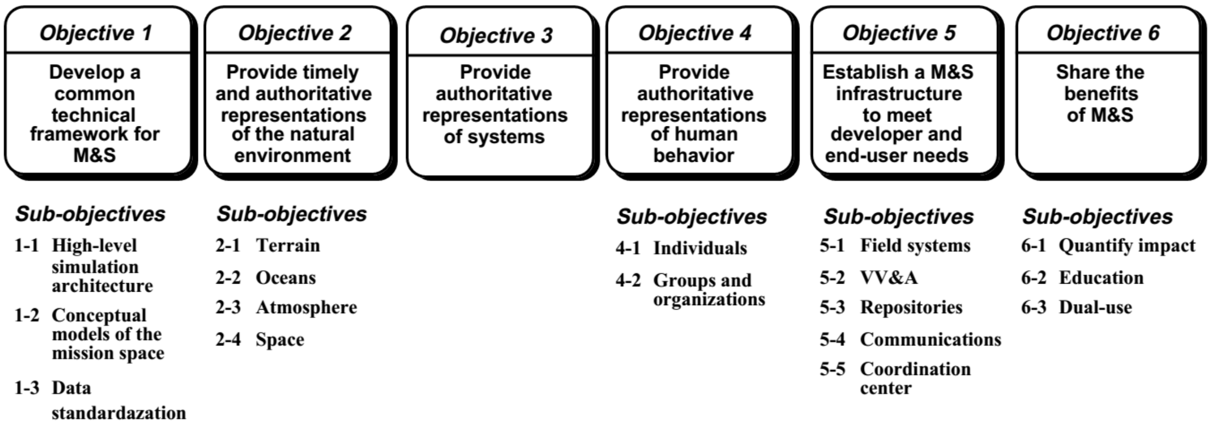


Figure 10: DoD M&S Objective and Sub-Objective

Source: MODELING AND SIMULATION (M&S) MASTER PLAN, 1995

2.1.2 Joint Live Virtual Constructive Data Translator (JLVCDT) Framework

W. Bizub et al. (2006) presented the Joint Live Virtual Constructive Data Translator (JLVCDT) Framework to provide interoperability for a seamless joint training environment. The JLVCDT is intended to provide equal or better functional capabilities than current translators, but in a more common, usable and open software architecture. This research suggested a harmonization of SSAs for the LVC community.

Cutts, Gustavson, and Ashe (2006) studied that Base Object Model (BOM) as a unifying approach to object modeling could provide an effective approach to converging Object Models across the multiple SSAs.

W. W. Bizub and Cutts (2007) described a plan for moving toward improved LVC interoperability based on the findings, and recommendations assimilated from the activities in the DoD M&S Steering Committee (SC) Live Virtual Constructive Way Ahead (LVCWA) study. U.S. DoD M&S SC sponsored study was established with the objective of developing an LVCWA. The study team is exploring and assessing a number of alternatives supporting simulation interoperability (at the technical level), business models, and the evolution process of standards management across the DoD. LVCWA study was to study the issues related to Live, Virtual and Constructive interoperability and to recommend a way ahead to increase interoperability across several areas: notional definition of the desired future SSA, the business models, and methods in which SSAs should be evolved and compliance evaluated. The LVCWA provided a blueprint for the new LVC SSA issues.

Gustavsson et al. (2009) presented use-case and interoperability issues that are needed to be considered when creating integration and interoperability methodology and applications to support Live, Virtual and Constructive simulation based on an operational need driven perspective. The authors focused on C2 LVC simulation and the correspondent interoperability issues on information integration rather than architecture and/or protocol Integration.

2.2 Interoperability, Integration and Composability

M&S communities have recognized the importance of LVC ***interoperability***, ***integration*** and ***composability*** for a seamless LVC simulation (Tolk, 2012). For successful LVC simulations, especially the importance of achieving interoperability of the simulation systems, integration of infrastructure and composability of the underlying combat models are being emphasized in the M&S as well as many other application areas. Interoperability, integration and composability have also been identified as the most technically challenged aspects of a U.S. Army LVC-IA since at least 1996.

Page, Briggs, and Tufarolo (2004) suggested distinguishing clearly between the three concepts for LVC simulation. Interoperability concerns the realm of the software implementation of the model (e.g. are the data types consistent; this includes exchange of data elements via interfaces, the use of middleware, mapping to common information exchange models, etc.). Integration concerns the physical/technical realms of connections between systems, which include hardware and firmware, protocols, networks, etc. Composability concerns the modeling part (e.g. two models are composable if their objectives and assumptions are properly aligned). LVC Architectures must holistically address all three aspects in well aligned systemic approaches.

2.2.1 Interoperability

The distributed simulation systems have some disadvantages. The issues most related to distributed simulation systems are interoperability concerns. When more than one SSA must be used in the same simulation environment, the SSA differences result in interoperability problems. A lot of additional work has to be done after interconnection is ensured, to reach higher levels of interoperability (semantic interoperability as shown in Figure 11). The study of interoperability concerns methodologies to be interoperable between different simulation systems distributed over a network system.

2.1.1.1 Definition of Interoperability

These example definitions of interoperability from the literature illustrate the variations that can be found:

According to DoD, *M&S interoperability* is defined as the ability of a model or simulation to provide services to, and accept services from, other models and simulations, and to use the exchanged services to enable them to operate effectively together (DoD, 1995).

Dumanoir, Parrish, and Sotomayor (2007) defined *LVC interoperability* as the ability for assets, models, and effects from one training environment to be seen, affect, and be affected within the rest of the training environment. According to the North Atlantic Treaty Organization (NATO) Modelling and Simulation Standards Profile (NATO, 2009):

> “Definition of *interoperability* among simulations is that the capability for simulations to physically interconnect, to provide (and receive) services to (and from) other simulations, to use these exchanged services in order to effectively work together. This definition refers mainly to *technical interoperability* which means the possibility to physically interconnect then communicate”.

The RTI Interoperability Study Group proposes that the following definition of Interoperability be adopted by Simulation Interoperability Standards Organization (SISO) (Myjak, Clark, & Lake, 1999):

> "Interoperability means there is functional equivalence provided by interchangeable components within a system or process in order to allow its components to be able to work together with no prior agreement over an agreed-upon data communications path."

Therefore, we can clearly classify if one simulation system is interoperable or not. For example, if two simulation systems are stand-alone and are not connected to supporting networks and other infrastructure elements, they obviously cannot exchange anything. So, we can say there is no interoperability.

2.1.1.2 Levels of Conceptual Interoperability Model (LCIM)

The Levels of Conceptual Interoperability Model (LCIM) was developed to cope with the different layers of interoperability of modeling & simulation applications. Since its first introduction by Tolk and Muguira in 2003 (Tolk & Muguira, 2003), the LCIM has evolved. The current version of LCIM is seven layers that are a) no interoperability, b) technical interoperability, c) syntactic interoperability, d) semantic interoperability, e) pragmatic interoperability, f) dynamic interoperability, and g) conceptual interoperability as described as summarized in the Table 3 and Figure 11 (Tolk, 2012; Wang, Tolk, & Wang, 2009).

Table 3: Implications of LCIM

Level	Layer Name	Premise	Information defined	Contents clearly defined	Domain	Focus	Capability
L6	Conceptual	Common conceptual model	Assumptions , constrains etc.	Documented conceptual model	Modeling abstraction	Composability	High
L5	Dynamic	Common execution model	Effect of data	Effect of information exchanged			
L4	Pragmatic	Common workflow model	Use of data	Context of information exchanged	Simulation implementation	Interoperability	Medium
L3	Semantic	Common reference model	Meaning of data	Content of information exchanged			
L2	Syntactic	Common data structure	Structured data	Format of information exchanged			
L1	Technical	Common communication protocol	Bits and bytes	Symbols of information exchanged	Network connectivity	Integratability	Low
L0	No	No connection	NA	NA			

The figure also shows the area of integration (or integratability), interoperability, and composability together.

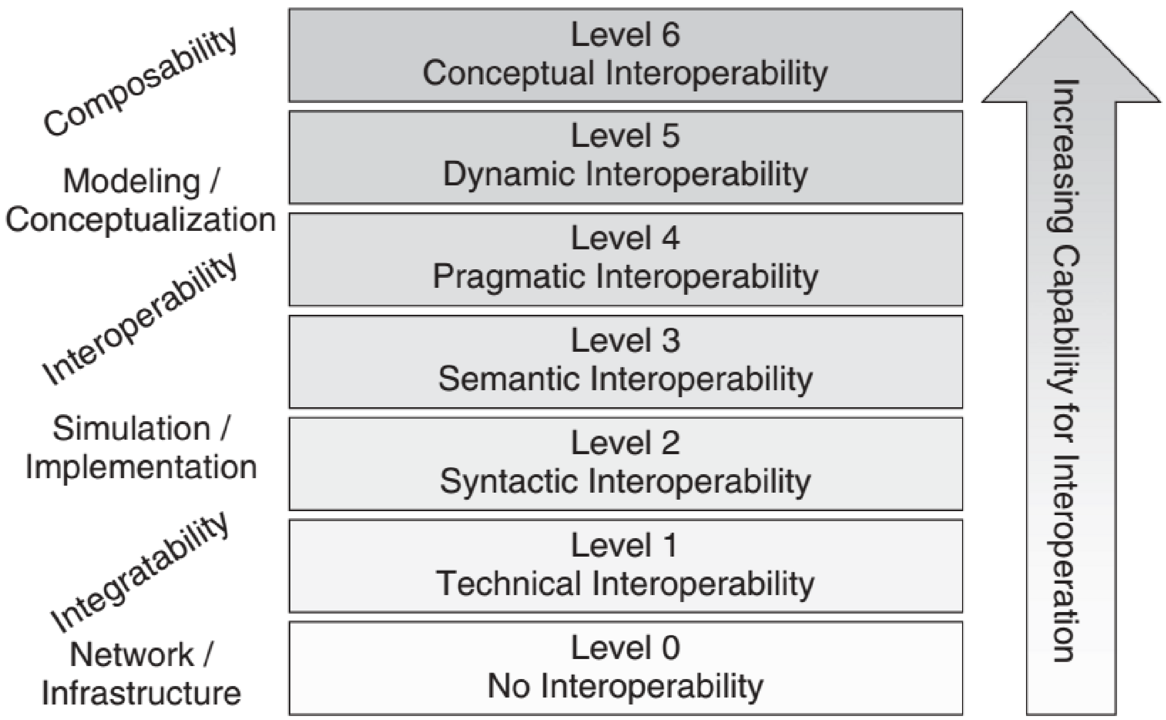


Figure 11: Levels of Conceptual Interoperability Model

Source: Tolk (2012)

The different levels can be characterized as follows (Andreas et al., 2007; Tolk, 2012):

● Level 0: Stand-alone system. There is *No Interoperability*.

● Level 1: On the level of *Technical Interoperability*, communication protocol (or infrastructure is established for exchanging information and data between simulation systems. The underlying networks and protocols are unambiguously defined. This level supports integratability.

● Level 2: The *Syntactic Interoperability* level introduces a common structure to exchange common data format. On this level, a common protocol to interpret and structure the data is used; the format of the information exchange is unambiguously defined. This level belongs to the domain of interoperability.

● Level 3: The level of *Semantic Interoperability* can be obtained, if a common information is exchanged by introducing a common terminology. On this level, the meaning of data is shared and the pieces of information that can be composed to objects, messages, and other higher structures are identified using common terms to address these structures.

● Level 4: The *Pragmatic Interoperability* recognizes the pattern (or methods and procedures) in which data are organized for the information exchange, which are in particular, the inputs and outputs of procedures and methods to be called. This is the context in which data are exchanged as applicable information.

● Level 5: On the level of *Dynamic Interoperability*, as a simulation system operates on data over time, the state of that simulation systems will change. This level recognizes various simulation system states.

● Level 6: Finally, if the conceptual model is aligned, the highest level of interoperability is reached: *Conceptual Interoperability*. The conceptual model means the assumptions and constraints of the meaningful abstraction of reality. This level requires that conceptual models will be documented based on engineering methods enabling their interpretation and evaluation by other engineers. The conceptual model is described in detail later in Section 2.4.

2.1.1.3 Interoperability Inhibitors

W. W. Bizub and Cutts (2007) investigated several key inhibitors to Live, Virtual and Constructive (LVC) simulation interoperability.

2.1.1.3.1 Lack of Understanding of the Interoperability Issues between Live Virtual and/or Constructive

If M&S community wants a seamless LVC interoperability, the differences and features between Live, Virtual and Constructive simulation environments must be thoroughly investigated and documented.

2.1.1.3.2 Differences in Intended Use

As mentioned earlier, the multiple SSAs were developed for different domains and the particular needs of each community.

2.1.1.3.3 Incompatibilities in Data Transfer/Object Modeling between SSAs

Data transfer/object modeling has steadily been a problem to interoperability and composability, even within a single SSA. This means that common and standard object modeling referential is required to ensure a seamless LVC simulation.

2.1.1.3.4 Lack of Composability

The composability is intended to enable effective integration, interoperability, and reuse. However, the composability across the M&S community has not adequately been achieved. For example, the deficiency of composability inhibits the ability to achieve interoperability between the HLA and TENA (Cutts et al., 2006; Rieger & Lewis, 2006). Therefore, a single object modeling methodology, focused on achieving composability, must be considered in the LVC Architecture Way Ahead (LVCWA) study.

2.1.1.3.5 Systems Engineering Process

Each SSA has various processes and does not address each other's domain. A single system engineering approach is desirable and would be a significant enabler for LVC interoperability.

2.1.1.3.6 Business Process Attributes

Each SSA adopted different business strategies for governance and implementation. For instance, DIS and HLA is international standards based on a commercial off the shelf (COTS) implementation strategy, whereas ALSP, CTIA and TENA are an adopted Government off the shelf (GOTS) solution that emphasizes development and control by a U.S. Government agency and open access to "their" community of interest.

2.1.1.3.7 Middleware / Infrastructure Incompatibility

Although the SSAs provide well-defined user Application Programmers Interface (API) and a set of services to distribute data between producers and consumers, they have chosen different strategies depending on the intended usage.

2.2.2 Integration

The terms *Interoperability* and *Integration* are often used interchangeably by some, which might create confusion. It is necessary to clarify the differences and similarities. While *Interoperability* is a property (or quality) of integration that ensures a level of independence between existing and future systems or organizations, *Integration* is the process of linking together diverse systems or organizations (Dumanoir, 2012). According to Petty and Weisel (2003), integration is the process of configuring and modifying a set of components to make them interoperable and possibly composable. Integration creates *network-centric linkages* to collect, retrieve and exchange data among Live instrumentation, Virtual simulators and Constructive simulations as well as between the joint military and specific service command systems. Integration also bridges together data management, exercise management, exercise collaboration and updating training support systems. Therefore, the more the process of linking LVC simulations through a suitable technology or protocol is developed, the more simulation interoperability will be exploited within a federated simulation environment.

2.2.3 Composability

According to DoD (1995), Composability is defined within the DoD M&S Master Plan as "the ability to select rapidly and assemble components to construct meaningful simulation systems to satisfy specific user requirements." Such composability is intended to "enable effective integration, interoperability, and reuse." The defining characteristic that distinguishes composability from interoperability is the ability to combine and recombine components into different simulation systems for different purposes (Benali & Saoud, 2010).

2.3 Comparison of Standard Simulation Architectures (SSAs)

In the U.S. DoD, the SSAs that have contributed to LVC simulation environments are SIMNET, ALSP, DIS, HLA, TENA and CTIA. These SSAs are commonly used and developed to meet the interoperability needs of distributed simulation.

2.3.1 SIMulation NETworking (SIMNET)

In 1983, the SIMulation NETworking (SIMNET) project was initiated by the Advanced Research Projects Agency (ARPA, at that time the Defense Advanced Research Projects Agency (DARPA)), with substantial support from the U.S. Army (Calvin et al., 1993). Thus, SIMNET became the first successful SSA of a large-scale, real-time, human-in-the-loop simulator networking for team training and mission rehearsal in the military. The intent of SIMNET architecture was for and used by the U.S. Army to support real-time distributed battlefield tank simulators of the Combined Arms Tactical Training System (CATT) to enable tank crews to operate side-by-side in a virtual training environment. The most dramatic feature of SIMNET that differentiated it from previous military simulators was the capability to have many objects playing together in the same Virtual battlefield. During an exercise, each Virtual simulator sends messages via the LAN to the other simulators to deliver information that they need to know about its appearance and actions. Each virtual simulator also receives, interprets, and responds properly to the messages received from the other Virtual simulators (Calvin et al., 1993). SIMNET realized over 250 networked simulators at 11 sites in 1990 (Fujimoto, 1999).

The success of the SIMNET led to the incorporation of all its essential elements into the Distributed Interactive Simulation (DIS) standard. As a result, the SIMNET architecture was confirmed that distributed, interactive simulations are effective in the Virtual world.

2.3.2 Aggregate Level Simulation Protocol (ALSP)

In the early 1990s, soon after the inception of the SIMNET project, The ARPA recognized the need to connect aggregate level combat simulation systems (or war games) and focused on faster than real time simulation. The ARPA was searching for an alternative method for synchronizing distributed aggregate level combat simulation systems to provide for a theatre-level experience for battle-staff training. The Aggregate Level Simulation Protocol (ALSP) under the auspices of Advanced Distributed Simulation (ADS), provided a mechanism for the integration of existing simulation models to support training via theater-level simulation exercises (Weatherly et al., 1996). ALSP enabled war game simulation systems from the Army, Air Force, and Navy, for example, to be brought together in a single exercise to analyze joint military operations. ALSP used synchronization protocols for analytic simulation system.

2.3.3 Distributed Interactive Simulation (DIS)

In the early 1990s, Distributed Interactive Simulation (DIS) standard architecture using the technical principles introduced by the SIMNET project was created to support virtual battles involving Semi-Automated Forces (SAF) (Jeffrey S Steinman & Hardy, 2004). DIS was standardized as IEEE 1278. The DIS used Protocol Data Units (PDUs) which used standard messages exchanged to convey to the state about entities and events. The PDUs were comprised of object data related to a common function. All communications about simulation entities and their interactions occurred via the PDUs (Tolk, 2012).

From a distributed system viewpoint, DIS is truly plug and play and does not require any middleware. DIS does not require any additional software, so it is easy to use out-of-the-box. However, DIS is best only for training and exercises on LAN because of the potential for high latency in WAN. In addition, since PDUs are broadcast to all simulation systems on the network (or exercise, exercise is the DIS term for a one or more interacting simulation systems. Compare to federation in HLA), bandwidth and computing resources can be consumed processing data that is not relevant to a specific simulation system. DIS requires that entities send a complete state update (heartbeat) at regular intervals (typically every 5 seconds) even if their state has not changed. In large scenarios, this can flood the network with update messages, which can result in dropped packets.

From a time management perspective, the DIS simulation system does not support time management and data distribution management. DIS supports only real time and no fast or slow simulation execution.

The DIS is still successfully used and supported by a large user community, but the DIS has since been replaced by the High Level Architecture (HLA) that expanded this DIS approach to include war game simulation systems the ALSP supports.

2.3.4 High Level Architecture (HLA)

The High Level Architecture (HLA) is the current leading SSA. In 1996, HLA was successfully developed by the U.S. DoD to promote interoperability and reusability between the many different types of simulations executing in distributed simulation environments (Jeffrey S Steinman & Hardy, 2004). HLA 1.3 became IEEE 1516 standard in 2001, then HLA Evolved came up with benefits of Modular Federation Object Model (FOM)/Simulation Object Model (SOM) in 2010. HLA has been adopted by NATO as well (Standardization Agreement (STANAG) 4603).

The HLA was designed to support two disparate applications of DIS and ALSP and to supplant both of them. In other words, the intent of HLA development was to combine the best features of DIS and ALSP into a single architecture that could also support uses in the analysis and acquisition communities while continuing to support training applications. Therefore, the HLA is a simulation architecture that enables several simulation systems to work together.

In a simulation based on HLA, federation, which consists of several interactional simulation systems, is a distributed simulation system that is used to realize a given simulation purpose. The simulation systems, application programs and components engaged in federation are called federates. In a federation, there are different kinds of federates as shown in Figure 12.

The HLA SSA is defined by three components (Dahmann, Kuhl, & Weatherly, 1998; Tolk, 2012):

● *HLA Rules* describe that simulation systems must obey to be compliant to the standard.

● *Object Model Template (OMT)* specifies what information is communicated between federates and how it is documented.

● *Interface Specification* is the specification of the interface between federates and the Runtime Infrastructure (RTI).

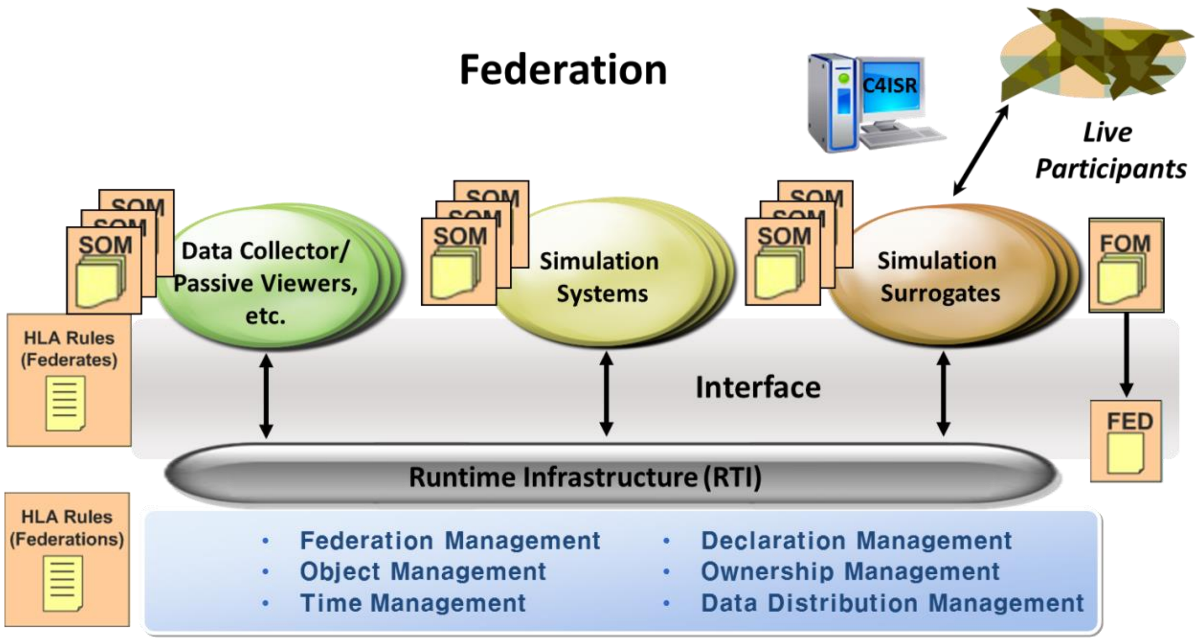


Figure 12: RTI and applications in the HLA

The Features of HLA:

● Each federate has a Simulation Object Model (SOM) that defines the data to be shared with other federates allowing reuse in different federations.

● Federation has a common Federation Object Model (FOM).

● Time Management can be used to ensure the correct ordering of events.

From the distributed simulation perspective, the HLA is based on the idea of separating the functionality of simulations from the infrastructure required for communication between simulation systems. This separation is accomplished by a distributed operating system called the RTI (Tolk, 2012). However, as HLA is tied to FOM, it is not truly plug and play.

From a communications perspective, the DIS broadcasting information to all simulation systems has serious implications on performance. On the other hand, the HLA allows individual simulation systems to filter data it wants to receive at many different levels via RTI (Tolk, 2012). This approach maximizes network performance and data distribution management makes it suitable for WAN environment. However, loosely coupled federation can encounter conceptual modeling issue, making it extremely difficult and results in a lot of verification cost for the simulation result (Jeffrey S. Steinman, 2013).

From a time management perspective, the HLA does include time management services to support event ordering. Support for time management allows to run simulation fast or slow as well. Global time advance and event ordering is implemented by means of synchronization algorithms (Tolk, 2012).

2.3.4.1 Run Time Infrastructure (RTI)

To use HLA, user must install an HLA RTI. The RTI is a software library that implements the HLA 1.3, 1516 and 1516-2010 (HLA Evolved) interface specifications as a fundamental component of HLA. An RTI is required to run applications using the HLA. The function of the RTI is to manage exchange of data between federates in a federation and provides information, synchronization, coordination and the HLA services. There are available RTIs software in the market like a MÄ K RTI by MÄ K Technologies and Pitch pRTI by Pitch Technologies, etc. However, RTIs from different vendors are functionality neither compatible nor interoperable with one another. The result is the adoption of a specific RTI, produced by a specific vendor, often for only a limited set of platforms.

2.3.5 Test and Training Enabling Architecture (TENA)

In the late 1990s, after the HLA initiative was in progress, the Test and Training Enabling Architecture (TENA) emerged. Currently, TENA is a SSA mainly used by the Office of the Secretary of Defense (OSD) Central Test and Evaluation Investment Program (CTEIP) to integrate testing, training, simulation, and high-performance computing technologies distributed across many facilities. TENA is also the Joint National Training Capability (JNTC) architecture for live training and is used primarily as communication architecture (PEO-STRI, 2006a). The TENA provides the architecture and the software implementation necessary to do three things. First, TENA quickly and economically enables interoperability among range systems, facilities, simulation systems, and C4ISR systems. Second, TENA also promotes reuse for range assets utilization and for future developments. Lastly, TENA provides composability to assemble, initialize, test, and execute a system rapidly from a pool of reusable, interoperable elements (Tolk, 2012). The goals of the TENA Software Development Activity (TENA-SDA) are to enable interoperability among U.S. DoD testing and training ranges, facilities, and simulations quickly and cost-effectively, and to foster reuse of range assets (Noseworthy, 2008). Figure 13 depicts TENA architecture.

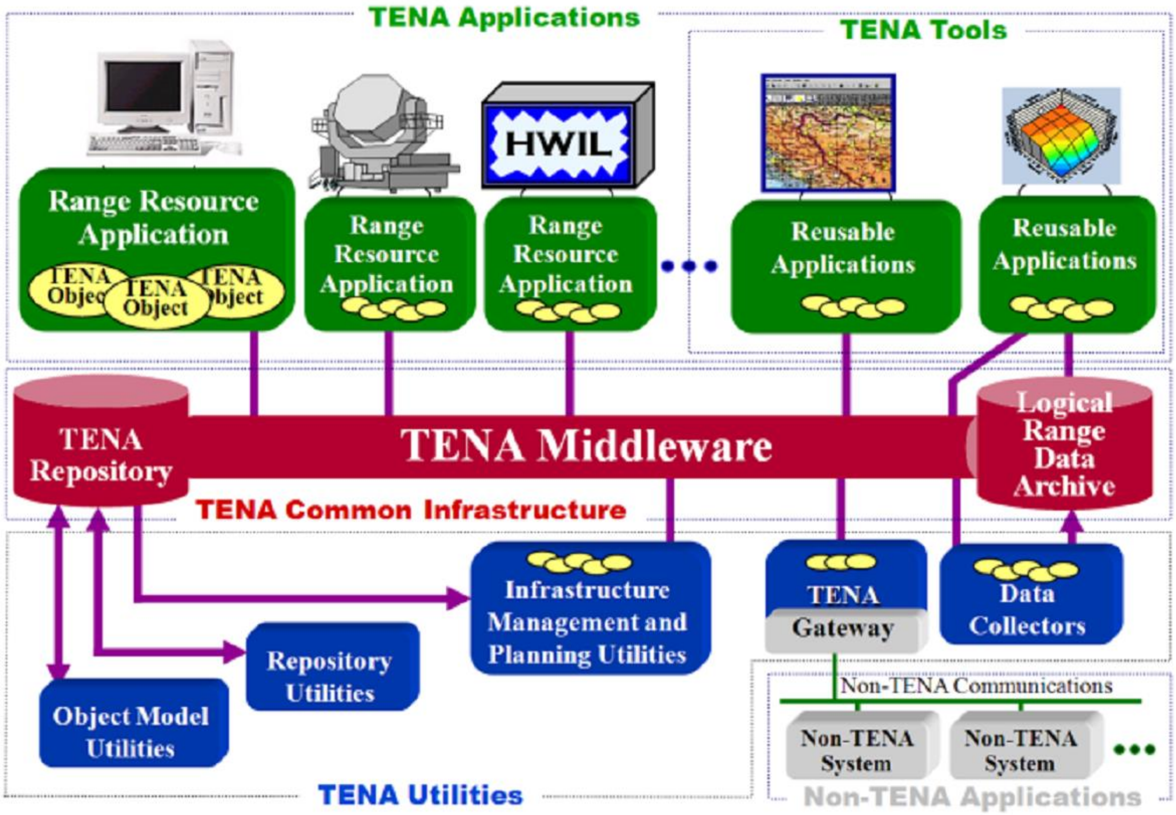


Figure 13: TENA Architecture

Source: Noseworthy (2008)

The core of TENA is the TENA Common Infrastructure, including the TENA Middleware, the TENA Repository, and the TENA Logical Range Data Archive. There is also the TENA Object Model, which defines the common data and interfaces shared by all range applications. In addition, there are a number of tools, utilities, and gateways to enable many range resources located at geographically dispersed ranges to be integrated together in a timely manner (PEO-STRI, 2006a).

From a distributed systems view, TENA separates the functionality of range assets from the infrastructure required to communicate among assets using middleware. The TENA

Middleware facilitates all data exchange and control commands between range systems. More importantly, the TENA Middleware provides range system developers with a unified API to support the real-time exchange of software objects, messages and data streams (PEO-STRI, 2006a).

On the other hand, TENA Repository contains all the information relevant to TENA that is not specific to a given test or training event. The TENA Repository is web-enabled and functions, in essence, as a large database of databases, allowing event planners to browse and select capabilities that can be easily configured and used to support an event. In addition, TENA Logical Range Data Archive: Stores and allows retrieval of all the persistent information associated with a test or training event (PEO-STRI, 2006a).

From a time management perspective, there is no requirement for time management to support event ordering because of the given nature of real-time range assets. This includes synchronization and time setting services, as well as maintaining a global clock for exercises (Tolk, 2012).

2.3.6 Common Training Instrumentation Architecture (CTIA)

The Common Training Instrumentation Architecture (CTIA) was developed to support the U.S. Army's Live Training Transformation (LT2) product line. The CTIA defines the framework for the design and development of common, reusable components that establish essential commonality across the family of LT2 systems. The CTIA establish the standards, interfaces and protocols that are the foundation upon which to build the family of composable, fully integrated LT2 training systems (Lanman, Becker, & Samper, 2009). The CTIA and LT2 were explained in detail in Section 2.7.2.

2.3.7 Comparison of SSAs

Figure 14 shows the relative use of SSAs as surveyed by the LVCAR study. Today, the most widely used LVC SSAs in the DoD are HLA, DIS, TENA and CTIA. HLA is the current leading SSA. The LVCAR survey presented that the ALSP has a usage under 5%, DIS 35%, HLA 35%, TENA 15%, CTIA 3% and other is roughly 7% (Gustavsson et al., 2009).

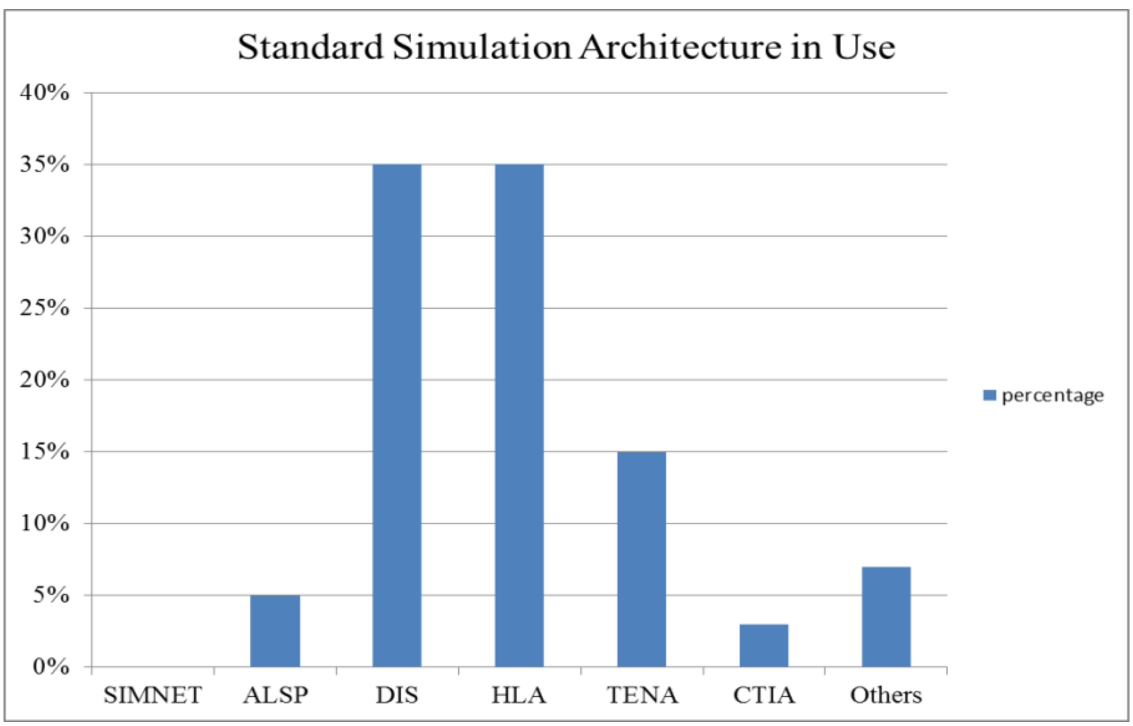


Figure 14: Usage Frequency of SSAs in the U.S.

2.3.8 Section Summary

Today, SSAs in use within the DoD have all been designed to meet the unique needs of one or more user communities as summarized in Table 4. Each SSA's execution data model and protocol have evolved and matured separately as an appropriate solution based on changing requirements. While the existence of diverse SSAs allows users to select the methodology that best meets their individual needs, these SSAs are not inherently technically interoperable because these separate evolutions have resulted in different methods for representing what is often similar information or phenomena. Therefore, the greatest need identified to be addressed is the interoperability between different SSAs. Incompatibilities between DIS, HLA, TENA and CTIA require the development of new single LVC SSA to effectively integrate the multiple existing SSAs into a single, unified set of simulation services. One benefit of having a common single LVC SSA is that models and simulation systems make use the same programming constructs and can therefore more easily interoperate (Jeffrey S. Steinman, 2013). The successful integration of LVC simulation systems might continue to rely upon the development of new single SSA. However, we concluded that migrating to a single LVC SSA was impractical in the near future. The simulation environments using various SSAs would be remained for the near future.

Table 4: Comparison of SSAs

	SIMNET	ALSP	DIS	HLA	TENA	CTIA
Organization		IWG	IEEE	AMG/IEEE/ SISO	AMT	PEO STRI
User Community	U.S.	U.S.	International	International	U.S.	U.S.
Business Model		GOTS	COTS	COTS	GOTS	GOTS
Level	Entity	Unit	Entity	Entity/Unit	Entity	Entity, Organization
LVC	Virtual	Constructive	Virtual	General	Range	Live asset
Time	Real time	Logical time	Real time	Real/Logical time	Real time	Real time
Percentage	0%	5%	35%	35%	15%	3%
Object Model			PDU	OMT	LROM	
Implementation			Plug & Play	RTI	TENA Middleware	
API		API		API	API	API

Note. AMG = Architecture Management Group, SISO = Simulation Interoperability Standards Organization, AMT = Architecture Management Team, IWG = Interface Working Group, PDU = Protocol Data Unit, OMT = Object Model Template, LROM = Logical Range Object Model, PEO-STRI = Program Executive Office for Simulation, Training and Instrumentation

2.4 Conceptual Model (CM)

What is a *Conceptual Model* (CM), and why is it important for LVC-ITA? Conceptual modeling is about abstracting a model from a real or proposed system. All simulation models are simplifications of reality (Zeigler, Praehofer, & Kim, 2000). However, the main issue is to abstract an appropriate simplification level of reality in conceptual modeling (Pidd, 2003).

In problem analysis and requirements analysis phase of simulation development, the CM can be used as a tool. The majority of the researchers consent that it is essential to develop CMs in the initial step of a simulation development life cycle (Pidd, 2003; Robinson, 2008).

According to Robinson, a conceptual model is defined as follows:

The CM is "a non-software specific description of a simulation model (that will be, is, or has been developed), describing the objectives, inputs, outputs, content, assumptions, and simplifications of the model" (Robinson, 2008). Therefore, the *Conceptual Modelling* is the process of creating the conceptual model. This definition is based on business-oriented simulation domain rather than the military domain. Robinson divided the simulation domain into two groups as military and business-oriented and describes the similarities, and differences between them. Robinson described that the military simulations often necessitate large-scale models developed by the development team. There is much interest in model reuse and distributed simulation whereas business-oriented simulations tend to be smaller in scale. In this context, the prime interest of this dissertation is in military simulation systems that include interaction, larger in scale and possibly distributed.

2.4.1 Modeling and Simulation (M&S) Process

This section describes some of the existing methods related with conceptual modeling. Federation Development and Execution Process (FEDEP), Synthetic Environment Development and Exploitation Process (SEDEP), Distributed Simulation Engineering and Execution Process (DSEEP), Conceptual Models of the Mission Space (CMMS) and Defense Conceptual Modeling Framework (DCMF) are introduced in brief and then compared to each other.

2.4.1.1 Federation Development and Execution Process (FEDEP)

The Federation Development and Execution Process (FEDEP), IEEE 1516.3, was developed as a guideline and recommended practice standard for developing interoperable HLA based federations. FEDEP is an overall framework overlay that can be used together with many other, commonly used development methodologies. However, there is one main concern from the M&S experts that the driving objective from user requirements was not emphasized in the FEDEP. On the highest level, FEDEP consists of the following seven steps as shown in Figure 15 (IEEE, 2003).

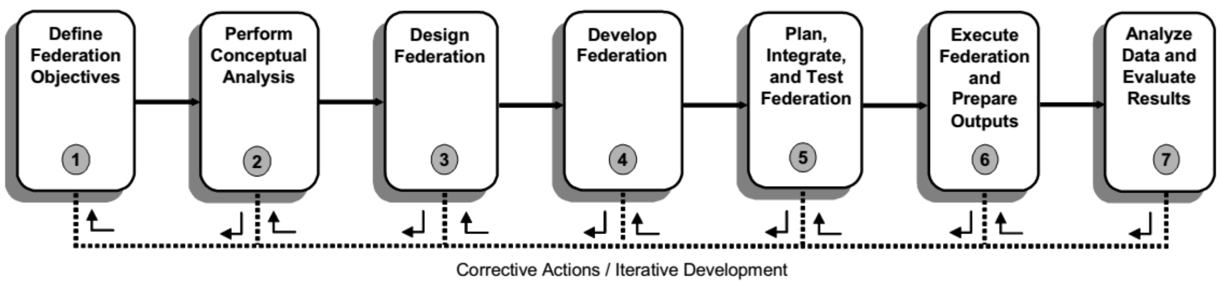


Figure 15: Top level process of the FEDEP

Source: IEEE (2003)

2.4.1.2 Synthetic Environment Development and Exploitation Process (SEDEP)

Using FEDEP as a starting point, SEDEP was developed. SEDEP improved the FEDEP, and added an additional process. On the top level, the SEDEP identifies the following eight steps including analysis user's needs as shown in Figure 16 (Ford, 2005). The SEDEP is frequently used in Europe for developments.

Figure 16: Top level process of the SEDEP

2.4.1.3 Distributed Simulation Engineering and Execution Process (DSEEP)

The generalization of FEDEP and SEDEP resulted in the latest standards, the Distributed Simulation Engineering and Execution Process (DSEEP). The FEDEP is still valid but has been succeeded by the IEEE 1730–2010 DSEEP. The purpose of the DSEEP is to describe a generalized process for building and executing distributed simulation environments as shown in Figure 17 (IEEE, 2011).

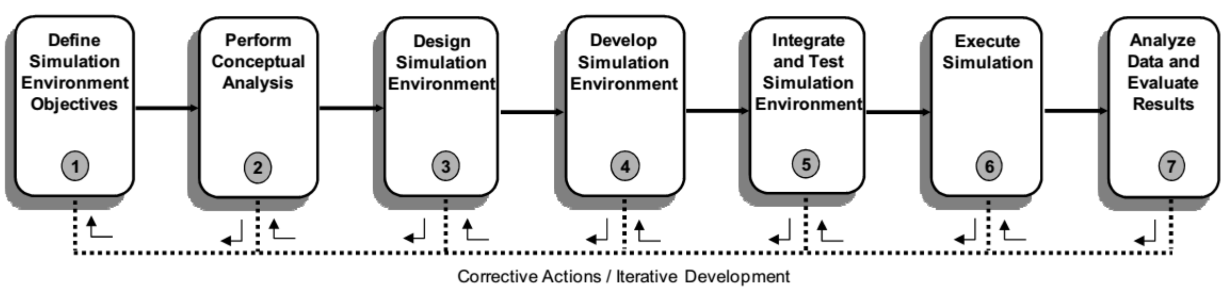


Figure 17: Top level process of DSEEP

Source: IEEE (2011)

The major steps and activities defined in the DSEEP are generally applicable to either single or multiple SSAs development. It describes a comprehensive set of technical issues that are either unique to multi-SSA development or are more difficult to resolve in multi-SSA simulation environments. Table 5 summarizes each step and comparison of FEDEP, SEDEP and DSEEP.

Table 5: Steps of FEDEP, SEDEP and DSEEP

Steps	FEDEP	SEDEP	DSEEP
Step 1	Define federation objectives	Analyze user's need	Define simulation environment objectives
Step 2	Perform conceptual analysis	Define federation user requirements	Perform conceptual analysis
Step 3	Design federation	Define federation system requirements	Design simulation environment
Step 4	Develop federation	Design federation	Develop simulation environment
Step 5	Plan, integrate, and test federation	Implement federation	Integrate and test simulation environment
Step 6	Execute federation and prepare outputs	Integrate and test federation	Execute simulation
Step 7	Analyze data and evaluate results	Operate federation	Analyze data and evaluate results
Step 8	•	Perform evaluation	•

2.4.2 Development of Conceptual Model of the CMMS and DCMF

2.4.2.1 Conceptual Model of the Mission Space (CMMS)

Although many M&S communities produced various framework definitions on conceptual modeling, they were with less guidance on the conceptual modeling phase. Additionally, increased use of modeling and simulation of military domain places a high demand for military knowledge management and how to use it. The main challenges are how to obtain, verify and keep the knowledge and method to accomplish this with minimum effort. Thus, in order to solve the issues associated with knowledge-based M&S, in 1995, the Conceptual Models of the Mission Space (CMMS) project originated by the U.S. DoD is one of the first initiatives providing detailed guidance on CM development activities.

Defense Modeling and Simulation Office (DMSO) extended CM definition and then introduced the term CMMS which can be defined as "simulation-implementation-independent functional descriptions of the real world processes, entities, and environment associated with a particular set of missions" (Sheehan et al., 1998).

The CMMS has the four principal components (Karagöz & Demirörs, 2011; Sheehan et al., 1998):

- Conceptual Mission Space Models: consistent functional descriptions of real-world military operations.
- Common Library: a database management system (DBMS) for model registration, storage, management, and release of conceptual models.
- Technical Framework: interoperability standards for knowledge acquisition and integration;

• a common syntax and semantics for describing the mission space

• a process definition for creating and maintaining conceptual models

• data interchange standards for integration and interoperability of mission space models

● Supporting Tools, Utilities and Guidelines.

2.4.2.2 Defense Conceptual Modeling Framework (DCMF)(Mojtahed, Lozano, Svan, Andersson, & Kabilan, 2005)

The Defense Conceptual Modeling Framework (DCMF) is the Swedish Defense Research Agency's (FOI) project for the development of CMs in the military domain. The FOI found the idea of CMMS concept very promising and then initiated the project to further study the CMs and improve the CMMS in the military context. The FOI developed DCMF from the original CMMS concepts by the U.S. DoD to make CMs applicable to many military scenarios without any loss of critical information. The DCMF's objective is to enhance interoperability, composability and reuse of knowledge for M&S. The final outputs from DCMF are the CMs that are called Mission Space Models (MSMs).

The DCMF process consists of four major phases as shown in Figure 18: Knowledge Acquisition, Knowledge Representation, Knowledge Modelling and Knowledge Use.

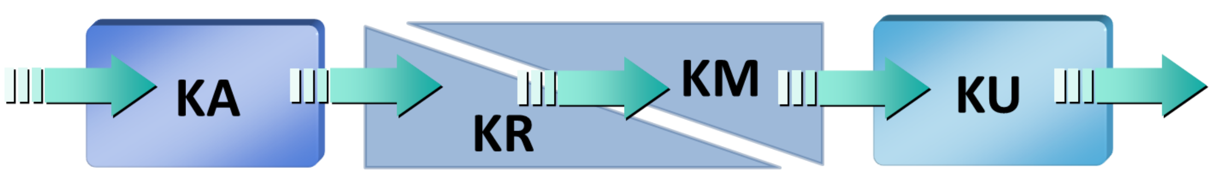


Figure 18: Four phases of the DCMF process

● *Knowledge Acquisition (KA)* is the acquiring phase which focuses on obtaining required information and knowledge from various sources. The important issues of the KA phase involve the limits of the area of the requirements scope, the identification of authorized knowledge sources, and the actual engineering.

● *Knowledge Representation (KR)* phase is to analyze, structure and formalize the acquired information. In this step, the human-readable and probably ambiguous information is transformed into a machine-readable and unambiguous format. The structuring and formalization of information should be performed in such a way that no information is lost in the process, and preferably so that the structured knowledge can be traced back to the source.

● *Knowledge Modeling (KM)* phase emphasizes the semantic analysis and modeling of the information. In this phase pragmatics is also an important part of the analysis and modeling. Another task of the KM is to merge the new CMs or components with the ones previously created.

● *Knowledge Use (KU)* is the final phase of the DCMF process involving the actual use of he modelled knowledge. In this phase the connection is strongest to the end user, and therefore, it is of great interest to visualize the acquired and modeled knowledge in different ways depending on the user's purpose and rights. To enable usage and reuse of that knowledge, it must be stored in a repository (i.e. DCMF Repository).

2.4.3 Section Summary

The DSEEP developed from FEDEP and SEDEP is recommended as the practice documents describing how to develop and execute a simulation environment. The DSEEP is unifying and single systems engineering process. The DCMF improved on the conceptual analysis of the FEDEP from the CMMS.

2.5 Bridging Solutions

This section provides solutions and its definition to bridge between simulation systems using heterogeneous SSA. When two or more different SSAs are used and need to be connected, when large-scale LVC simulation systems are integrated, or Simulation-to-C4I interoperability is demanded, in some cases, the current level of interoperability is attained through bridging such as the use of numerous a) Gateway, b) Middleware, c) Broker and d) Proxy, and e) Protocol. Myjak et al. (1999) also presented four different approaches to achieve interoperable solutions with HLA: the Gateway, Proxy, Broker, and Protocol solutions.

2.5.1 Gateway

Gateways are independent software applications that provide a connection and translation between two or more simulation systems that are supported by different SSAs (See Figure 19). A gateway focuses on the simulation systems, not the supporting SSA (Tolk, 2012). Currently, LVC interoperability can be achieved through gateway solutions, which can often restrict users to a limited set of capabilities that are common across the SSAs. A level of semantic interoperability is achieved through the use of numerous gateways to translate data sets among DIS, HLA, TENA, CTIA, ALSP, and other SSAs.

Though gateways are effective, the gateways represent another potential source of error (or failure) within the simulation systems, which can result in undesirable latencies into the simulation system, and increase the complexity of simulation systems. In addition, many gateways are legacy point solutions that provide support at most for an extremely limited number of services and only for very specific versions of the supported SSAs. Thus, it may be difficult to find a proper gateway that fully supports the needs of a given application. For the relatively small number of general-purpose gateways that are configurable, the effort required to perform the configuration function can be significant and can result in excessive consumption of project resources (APL, 2010).

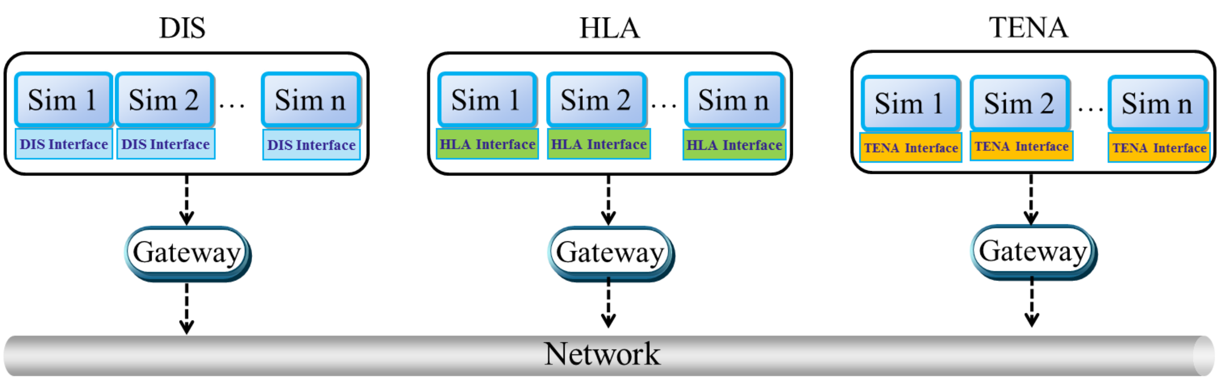


Figure 19: Gateway Configuration

2.5.2 Middleware

The use of middleware is a similar approach but provides the translation services in software directly coupled to the simulation instead of an independent application (See Figure 20). While middleware approaches are also effective, they introduce many of the same technical issues that are associated with gateways. In general, all of these "solutions" have limitations and cost implications that increase technical, cost, and schedule risk for multi-architecture developments (APL, 2010).

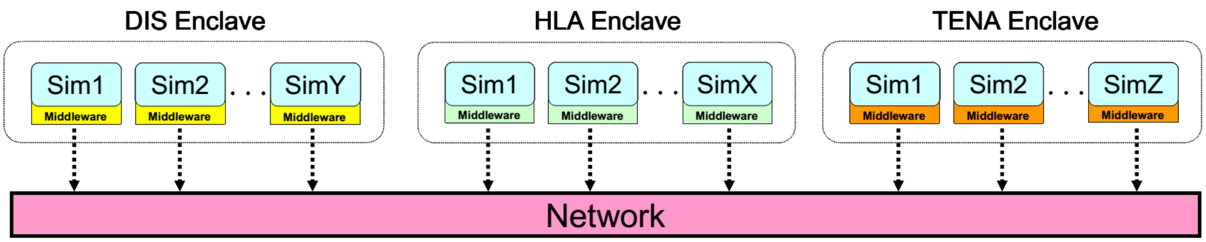


Figure 20: Middleware Configuration

Source: APL (2010)

2.5.3 Broker

A broker connects the SSAs with each other and allows the use of the services of one SSA to the other via interface program interfaces. Each broker translates between its native SSA. The translated data is translated again by the other brokers to their SSA. Simulation systems can interoperate with any other simulation systems for which a broker exists. The RTI broker provides the connection between RTIs in separate named federations and may have multiple connections to federations. Figure 21 shows the concept of broker.

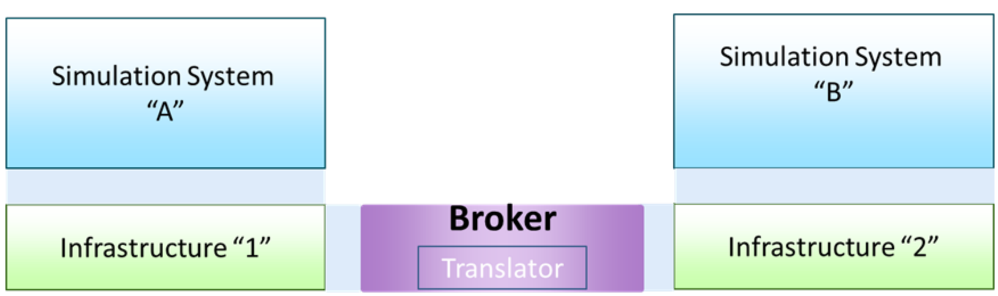


Figure 21: Brokers

2.5.4 Proxy

A proxy is a translation device that interconnects two different SSAs. It comprises the common elements such as entities and events that are shared between the two SSAs and uses the interface provided by the SSA for simulation systems (Tolk, 2012).

2.5.5 Protocol Solution

Protocol solutions extend the functionalities of the SSA on the network and protocol level down to binary level interoperability (Tolk, 2012).

2.5.6 Section Summary

This sectioned reviewed the bridging solutions that can be used when two or more different SSAs need to be connected. However, these solutions might result in undesirable latencies into the simulation system, and increase the complexity of simulation systems.

2.6 U.S. DoD Live-Virtual-Constructive Architecture Roadmap (LVCAR)

In April 2007, U.S. DoD Live Virtual Constructive Architecture Roadmap (LVCAR) study developed a recommended roadmap (way forward) regarding LVC interoperability to examine the differences among the major SSAs from technical, business, and standards perspectives and to develop a time-phased Set of Actions (SOAs) to improve interoperability within multi-architecture simulation environments in the future.

2.6.1 Purpose of the LVCAR

The first phase purpose of the LVCAR Study (or LVCAR Phase 1) was to develop a future vision and supporting strategy to achieve significant interoperability improvements in LVC simulation environments (Henninger et al., 2008). The second phase of this study (or LVCAR Phase 2) focused on the implementation of the recommended actions from the LVCAR Phase 1 Report. The LVCAR focused on four important dimensions of simulation interoperability: (a) technical architecture, (b) business models, (c) the standards evolution, and (d) management processes.

2.6.2 Main Four Fundamental Precepts of the LVCAR

In this section, the main four fundamental precepts are presented (Henninger et al., 2008).

2.6.2.1 Fundamental Precept #1: Do No Harm

The DoD should NOT take any immediate action to discontinue any of the existing SSAs. There is general consensus within the LVC user community that a long-term strategy based on architecture convergence would benefit the DoD. However, there are many design issues that must be resolved prior to implementing such a strategy, and that the actual implementation needs to be a well-planned, deliberate, evolutionary process to avoid adversely impacting participating user communities. Thus, near-term elimination of any existing SSA would be unwise. Rather, as the SSAs are gradually converged, the users themselves should decide if and when to merge their SSAs into some smaller set, based on both technical and business concerns. Any attempt by the

DoD to force a convergence solution on an unwilling user base is certain to meet strong resistance and likely to fail.

2.6.2.2 Fundamental Precept #2: Interoperability is NOT Free

The DoD must make the necessary investments to enable implementation of activities described in the LVC Roadmap. LVC interoperability is not free. It is not reasonable to expect that LVC interoperability goals can be met with little or no investment. Since the return on LVC investments is nearly impossible to accurately quantify in the near-term, it is understood that major new up-front investments are difficult to justify. The Roadmap will be designed to require only limited investment early in its implementation, with subsequent investments dependent on demonstrable progress. Without the necessary investments, the LVC Roadmap will be nothing more than a blueprint of what is possible to accomplish, with no mechanism to realize the associated benefits.

2.6.2.3 Fundamental Precept #3: Start with Small Steps

The DoD should take immediate action to improve interoperability among existing SSAs. The technical problems currently associated with the development and execution of mixed SSA LVC environments are well understood. They increase the technical risk and require more resources to address. While architecture convergence could reduce or eliminate several of these problems, it is not practical to expect any significant degree of convergence to occur for many years.

2.6.2.4 Fundamental Precept #4: Provide Centralized Management

The DoD must establish a centralized management structure for wide supervision of M&S resources and activities across developer and user organizations. Only a strong, centralized management team can prevent further divergence and make architecture convergence practical and effective. This team needs to have considerable influence on the organizations that own the existing SSAs, and must also have influence on funding decisions related to future LVC architecture development activities. Without centralized management, existing SSAs communities will continue to operate in line with their own self-interests, and the broader corporate needs of the DoD are likely to continue to be ignored.

2.6.3 Section Summary

To conclude, a key conclusion of the LVCAR effort was that evolving to a single SSA was impractical, and thus multi-architecture simulation environments would remain the state of the practice for the near future (APL, 2010). Thus, the best way forward is to enhance the interoperability of mixed-SSA events, while preserving options and positioning the community for some degree of SSA convergence in the future. This means that the best way forward is to take actions that can reduce or eliminate barriers to interoperability between existing SSAs.

2.7 U.S. Army Live Virtual Constructive-Integrating Architecture (LVC-IA)

In this section, in the area of architectural integration, we review the U.S. Army's overall Live Virtual Constructive-Integrating Architecture (LVC-IA). The LVC-IA is an effort and underlying architecture to support integration within and across Live, Virtual, and Constructive simulation-based training systems and operational C4ISR systems.

PEO STRI embraced the Product Line approach and it has been utilized to create a product line of interoperable products and services that maximize responsiveness to warfighter needs. Within PEO STRI there are product line initiatives within the Live, Virtual, and Constructive domains. These initiatives include the Live Training Transformation (LT2), Synthetic Environment Core (SE Core), and the Joint Land Component Constructive Training Capability (JLCCTC), as well as the Future Combat System (FCS) embedded training capability and the LVC-IA program.

Figure 22 provides a notional view of PEO STRI objective systems and their respective product lines and how they relate to LVC-IA, FCS and current and future BCS. The objective is for these PEO STRI product lines to be the key enablers of a Joint LVC-Training Environment (JLVC-TE) based on PEO STRI objective systems.

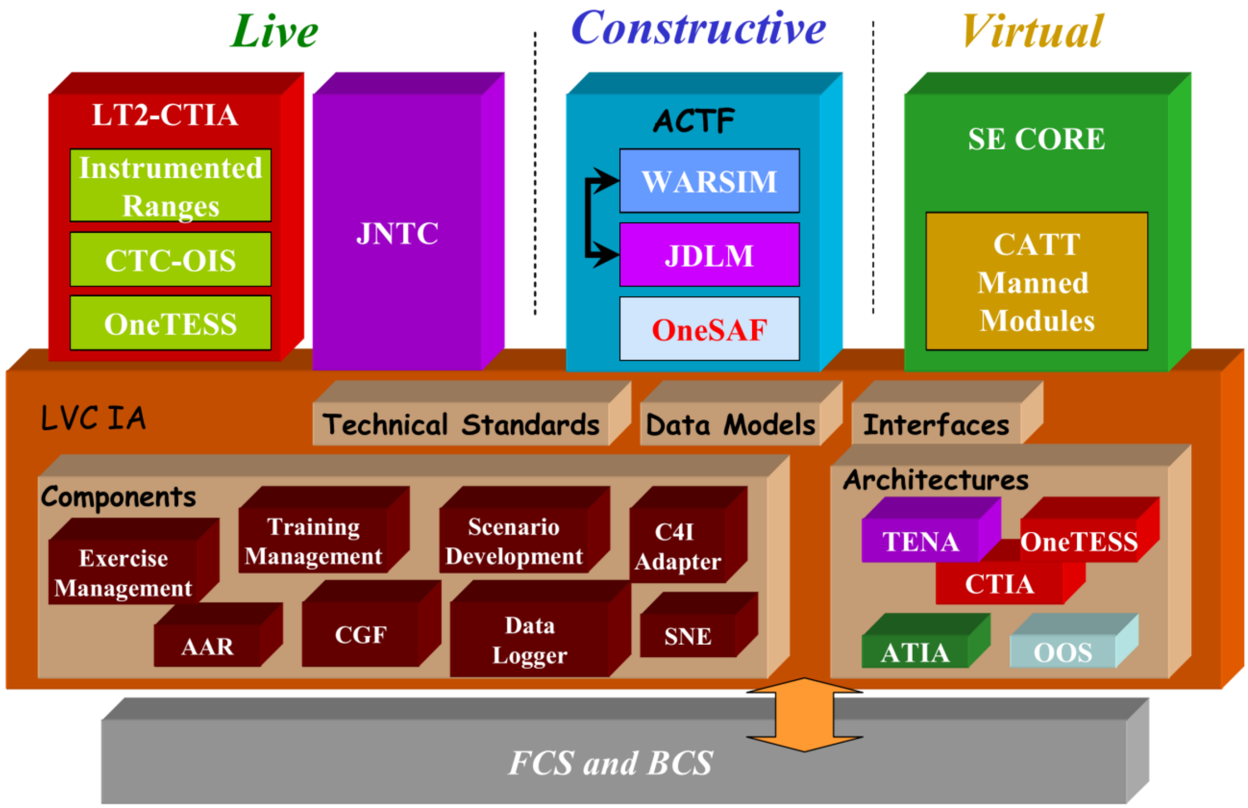


Figure 22: LVCTE Objective Systems

Source: Dumanoir, Pemberton, and Samper (2004)

2.7.1 Overview of LVC-IA

The U.S. Army LVC-IA project began in 2005. What is the U.S. Army LVC-IA? The LVC-IA is a set of protocols, specifications and standards that support a seamless and interoperable, integrated LVC environment where common hardware, software and network components and modules are interchangeable with other LVC components and BCS (Dumanoir, Keller, & Koenig, 2006; Dumanoir et al., 2004). In other words, the U.S. Army LVC-IA is a network-centric linkage that collects, retrieves and exchanges data among live instrumentation, virtual simulators, and constructive simulations as well as Joint and Army BCS (Rumpel & Vila, 2007; Shufelt Jr, 2006). According to Degnan (2009), LVC-IA is the aggregate representation of the foundational elements of the LVC Enterprise, including hardware, software, networks, databases and interfaces, policies, agreements, certifications/accreditations and business rules. LVC-IA is intrinsically an Enterprise Architecture, given the system-of-systems environment that it must support.

There are other associated terms related to LVC-IA (Degnan, 2009):

● LVC Enterprise: The overall enterprise of resources in which LVC activities take place.

● LVC Integration: The process of linking LVC simulations through a suitable technology or protocol to exploit simulation interoperability within a federated simulation environment such as the HLA.

2.7.2 Training Case based on LVC Simulation

Although current capabilities for integrating LVC training are limited, LVC simulation and operational C4ISR systems have been regularly integrated for some time in a limited number of settings. There are main challenges to integration of LVC simulation-based training events. One of them is the actual level of integration of Virtual into Live unit play have been limited because of the inherent lack of realism of having a Virtual simulation system engage a Live soldier or crew who cannot hear, see, or counter the Virtual system (Shanley, 2007). This section illustrates the known prior LVC exercise.

2.7.2.1 Integration of CCTT and JCATS in an LVC Exercise (Johnson et al., 2004)

The Joint Training Experimentation Program (JTEP) is a National Guard Bureau Project by the California National Guard (CANG). It is a multiphase, multiyear effort to develop a distributed training capability for the CANG that combines live, virtual, and constructive (LVC) simulations to support multi-echelon training. The Guard uses advanced live, virtual, and/or constructive systems to support training, but each system is used standalone. JTEP is intended to bring to the Guard the benefits of integrating existing or readily available training environments, and to enable LVC interaction over non-dedicated WANs.

In December 2003, the second JTEP demonstration was a battalion-sized exercise that has 125 total live and simulated entities conducted at Camps Roberts and San Luis Obispo in California and was a complete LVC integration. This demonstration linked the Joint Combat and Tactical Simulation (JCATS) that is a constructive simulation as shown in Figure 23, the Close Combat Tactical Trainer (CCTT) that is a virtual simulator as shown in Figure 24 and 25, and the Deployable Force-on-Force Instrumented Range System (DFIRST), a live instrumented training system was located 45 miles north from JCATS and CCTT at Camp Robert.

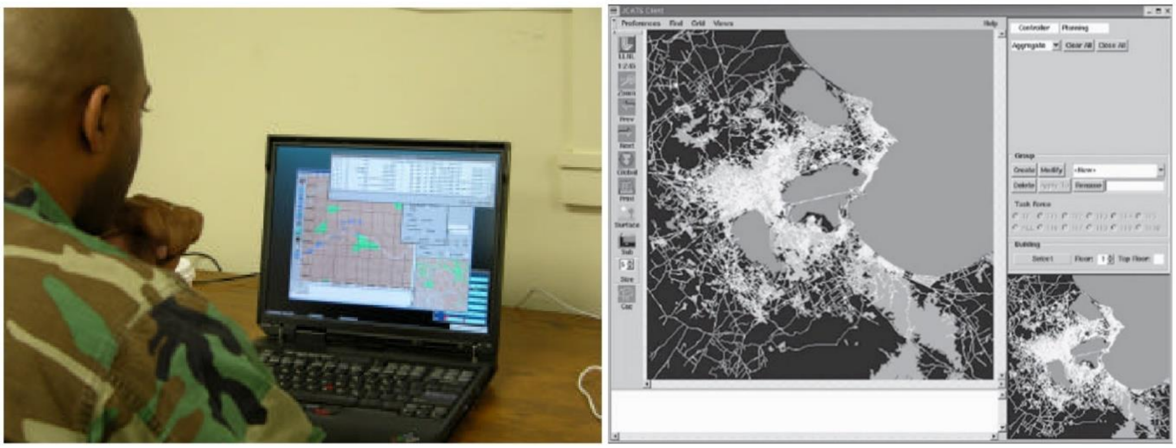


Figure 23: JCATS workstation and Display

Source: Johnson et al. (2004) and *JCATS: Simulation User's Guide* 2003)

Figure 24: Close Combat Tactical Trainer (CCTT)

Source: PEO-STRI (2013)
http://www.peostri.army.mil/PRODUCTS/CCTT/

Figure 25: California CCTT Mobile Units

Source: Johnson et al. (2004)

2.7.3 Training Concept of the U.S. Army LVC-IA

The current training environment consists of LVC simulations, simulators, and instrumentation systems that were not developed to interoperate with each other, nor link to BCS. However, the Army LVC-IA will support the Joint LVC-Training Environment (JLVC-TE) and the Joint National Training Capability (JNTC). The LVC-IA will facilitate increased unit competency in preparation for operating in a Joint, Interagency, Intergovernmental, and Multinational (JIIM) environment. The LVC-IA will enable a “plug-and-train” capability for units training in any domain or environment. The LVC-IA will rely on a robust communication network at home stations, CTCs and in operational environments. The JTEN, FTI, GIG and Warfighter Information Network-Tactical will provide the necessary bandwidth to move large packets of training data required for training and mission planning and rehearsals. Units will reach back to access large volumes of training data using standards and protocols developed by the LVC-IA into repositories developed by SE Core, LVC-IA, and FCS. Access to training support data will allow unit commanders to quickly develop scenarios using rapidly developed correlated TDBs resembling the mission area (geo-specific terrain) for training and mission planning and rehearsals anywhere in the world.

The U.S. Army LVC-IA operational view in Figure 26 shows the relationship between the LVC-IA, SE Core, LT2-FTS, JLCCTC, the other training environments and Net Enabled Command Capability (NECC) (Shufelt Jr, 2006).

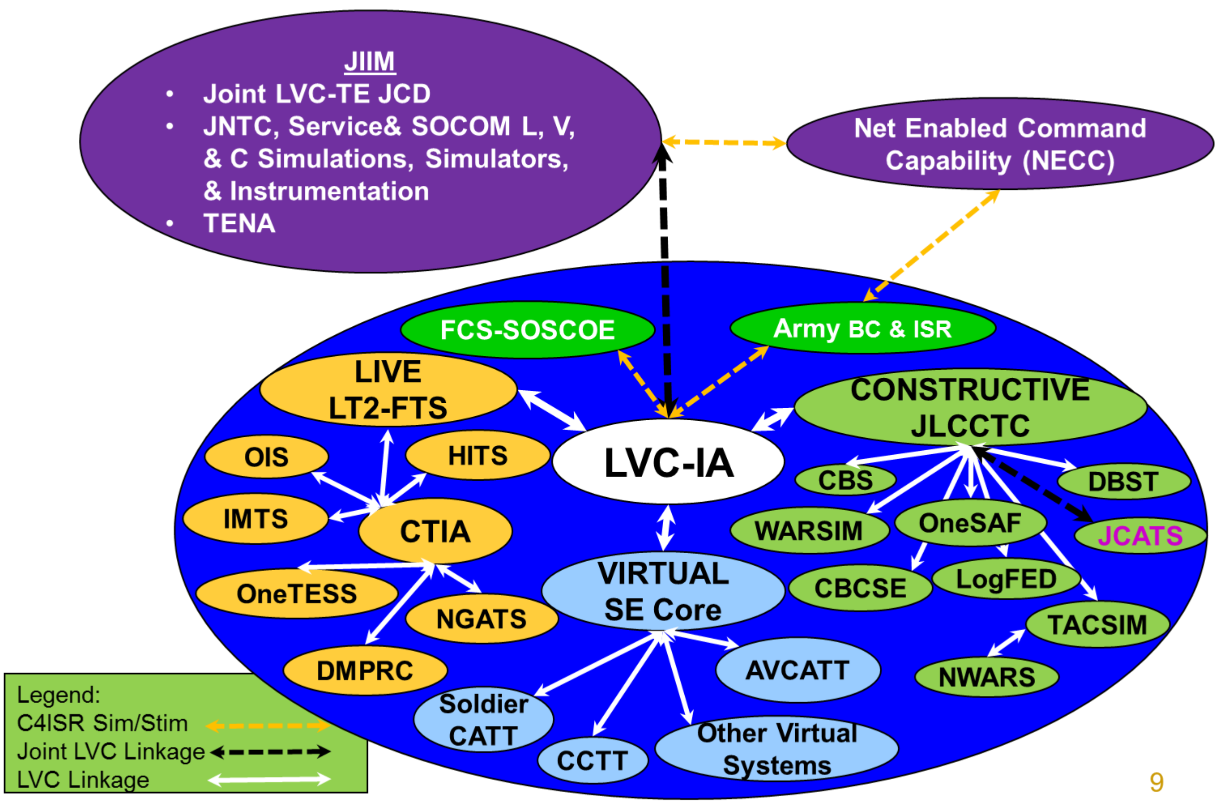


Figure 26: LVC-IA Operational View

Source: Shufelt Jr (2006)

2.7.4 U.S. Army LVC-IA Capabilities

The U.S. Army LVC-IA capabilities are briefly described below.

2.7.4.1 Scenario Generation & Initializing Exercise Preparation

The U.S. Army LVC-IA will provide an easy-to-use, composable exercise preparation toolkit that automates the capability to plan, design, prepare, and initialize a multi-echelon LVC exercise with detailed CGF. A LVC exercise preparation toolkit will enable the commander to quickly design and prepare an integrated LVC exercise reducing exercise preparation time increasing time available for training. This toolkit will allow system operators under the commander's guidance to reach into repositories of information to access exercise databases, scenarios, and other Army Training Information Architecture-Migrated (ATIA-M) information required to populate on-board embedded training systems, simulation systems and operational equipment.

2.7.4.2 Environmental Representations and Correlated Terrain Databases

The U.S. Army LVC-IA will provide a set of correlated and dynamic terrain models and standard algorithms. The terrain model must be interoperable with current and future force terrain services and address "fair fight" issues. Currently, LVC federates use different numerical systems to calculate simulated actions (e.g. line of sight, consumption, etc.) that involve digitized terrain. This method exacerbates terrain calculation when combined as a federation, as each of the respective federates has a different numerical system for interacting with the terrain. Correlated dynamic terrain models remove the need for translating or regenerating the terrain and supports efficient terrain calculations.

2.7.4.3 Data Collection and Specification

The U.S. LVC-IA will provide means to collect exercise data based on the commander's specified criteria to facilitate the conduct of In-Progress Reviews and AARs. A dynamic, automated data collection system based on specific criteria will enable commanders and leaders to objectively evaluate the training status of their crews, units and battle staffs.

2.7.4.4 In Progress and After Action Reviews

The LVC-IA provides a set of easy-to-use, multimedia data organization, presentation, and production capabilities required to assist in the development of in-progress review and AAR products, as well as teaching and training aids to assist in the facilitation of AAR. AAR production tools, teaching and training aids linked to all LVC components, embedded training systems, and operational equipment give commanders and leaders at all levels the ability to control their own exercises, provide immediate feedback, and reduce the need of high overhead support. An option being considered assumes the in-progress review or AAR data required for a live exercise, and is a super set of the data necessary for a constructive exercise.

2.7.4.5 Multi Directional Stimulation/Interaction of Operational & Training Equipment

During combat operations, the entire spectrum of information operations contributes to the generation and update of the Common Operating Picture (COP). The BCS constantly collects, collates and fuses inputs from various levels of command in order to provide commanders, battle staff, and soldiers with the information they need to execute their mission. During training and mission planning, preparation and rehearsal, the entire spectrum of information stimulus that contributes to a COP must also be present in order to facilitate battle-focused training.

The U.S. Army LVC-IA will fully stimulate and interact with joint and unit force BCS so commanders, leaders, and staff can fully interact with the battle command operational process and manipulate LVC components. In addition, the U.S. Army LVC-IA will simulate and emulate information exchange from other BCS. The U.S. Army LVC-IA will also provide linkages with on-board, embedded training systems when necessary and stimulate those systems with simulated and/or live data. The U.S. Army LVC-IA will also exchange data and services with Training Aids, Devices, Simulations, and Simulators (TADSS) systems, enabling the exchanged services to effectively operate together.

2.7.5 Components of the U.S. Army LVC-IA

The LVC-IA is the U.S. Army's very important initiative to integrate the future Live, Virtual, and Constructive simulation systems with operational C4ISR systems to support mission-rehearsal-type activities, as well as future training events.

Three major components of the LVC-IA are (a) Live Training Transformation – Family of Training Systems (LT2-FTS), (b) Synthetic Environment Core (SE Core), and (c) Joint Land Component Constructive Training Capability (JLCCTC) as shown in Figure 27.

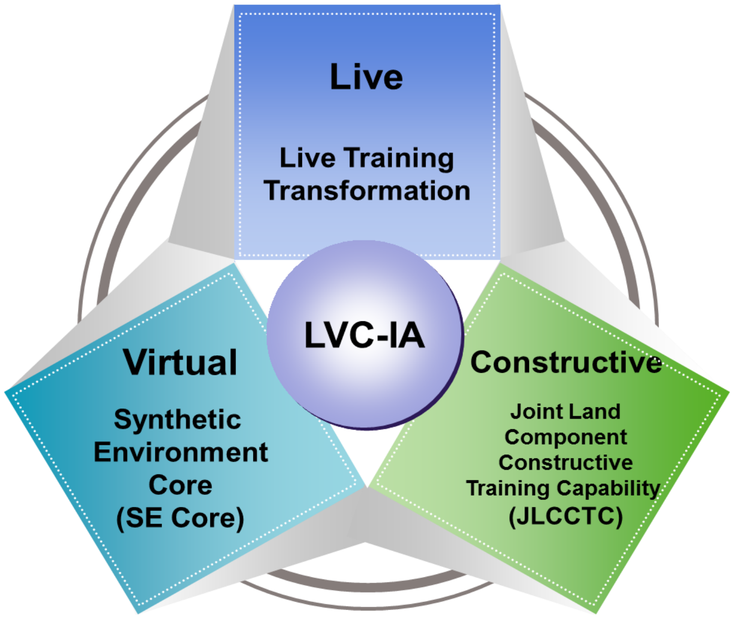


Figure 27: Three Major Components of the LVC-IA

2.7.5.1 Live Component: Live Training Transformation – Family of Training Systems (LT2-FTS)

This section describes the LT2-FTS which is Live component of LVC-IA.

2.7.5.1.1 Live Training Transformation (LT2)

The *Live Training Transformation (LT2)* is a strategy that takes advantage of the product line engineering development concepts and principles to guide the acquisition of the family of live training programs under the purview of the U.S. Army Program Executive Office (PEO) Simulation Training and Instrumentation (STRI), and Program Manager for Training Devices (PM TRADE) (Dumanoir & Rivera, 2005). The LT2 is the U.S. Army initiative to develop a *Live training range product line* that includes capabilities centered on a common architecture, known as the *CTIA*, and common plug-and-train components called *LT2* components, see Figure 28 (Dumanoir & Rivera, 2005; Rivera, Samper, & Clinger, 2007, 2008).

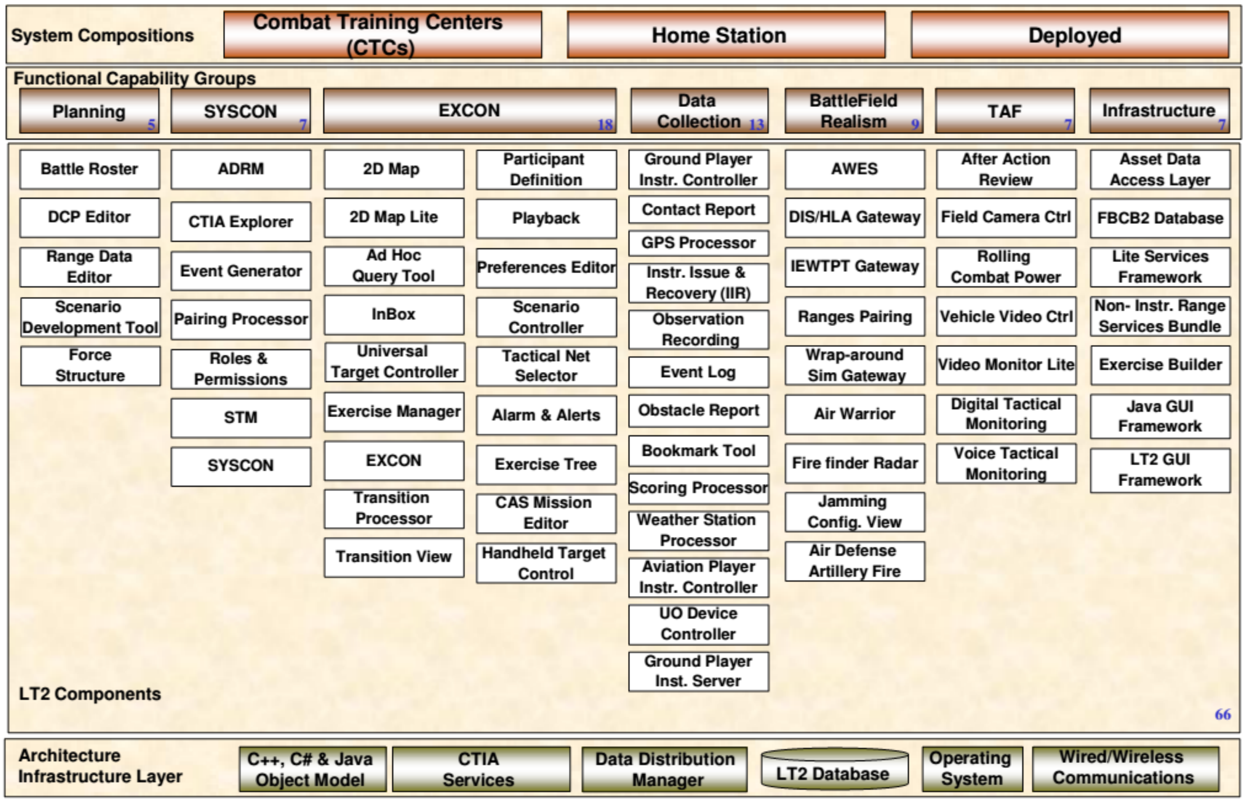


Figure 28: LT2 Component Product Line Framework

Source: Rivera et al. (2007)

The U.S. Army PEO STRI has established a LT2 product line approach to developing a Family of Training Systems (FTS) that provide the ground maneuver training range functions supporting Army Live and Joint training environments (Rivera et al., 2008). The LT2 product line strategy is required to synergize training instrumentation, targets, and tactical engagement simulation systems to ensure the efficiency and effectiveness of training during peacetime, mobilization, mission rehearsal, and in-theatre during deployed military operations (Dumanoir &

Rivera, 2005; Rivera et al., 2007, 2008). LT2 training systems will also provide interfaces to Virtual and Constructive training domain systems, the Army's C4ISR infrastructure systems, Future Combat System (FCS) platforms, and to components of the Joint National Training Capability (JNTC). LT2 products are constructed using a "family of components" approach, that maximizes software reuse, provides common functionality, and ensures hardware and interfaces performance and standards (Dumanoir & Rivera, 2005; Rivera et al., 2008).

The product types included in the LT2 live training domain are as follows (Dumanoir et al., 2004; Dumanoir & Rivera, 2005):

- Combat Training Center (CTC) - Objective Instrumentation Systems (OIS).
- Homestation Instrumented Training Systems (HITS).
- Integrated - Military Operations in Urban Terrain (MOUT) Training Systems (I-MTS).
- Instrumented Ranges which include Digital Multi-Purpose Range Complexes (DMPRC), Digital Multipurpose Purpose Training Range (DMPTR), and Battle Area Complex (BAX).

Through success of the product line strategy, LT2 will provide a common set of components that provides an integrated and interoperable training solutions for Live collective training.

2.7.5.1.2 LT2 Product Line Management Concept of Operations (PLM CONOPS)

The LT2 product line is implemented and managed as described in the LT2 Product Line Management Concept of Operations (PLM CONOPS). To maximize commonality and reuse of component and to ensure interoperability, the LT2 PLM CONOPS focuses on the overall requirements of all live domain training systems, with the LT2 strategy objectives to reduce fielding time, minimize programmatic costs, and enhance training benefits afforded to the soldier. The purpose of the LT2 CONOPS is to delineate the implementation and management processes necessary to provide oversight and coordination during the definition, development, and sustainment of the LT2 product line products, and its architecture and components.

This CONOPS also describes the processes, methods, roles and responsibilities, and tools required to manage the LT2 product line. This CONOPS establishes the PM TRADE management structure and processes required to execute the LT2 strategy across all PM TRADE programs, and the new live training capabilities defined in the approved LT2-FTS Initial Capability Document (ICD).

2.7.5.1.3 LT2 Family of Training Systems (LT2-FTS)

The LT2 strategy addresses a set of operational requirements defined by the approved eight existing live training Operational Requirements Documents (ORDs), and is being transformed into an Army program as a Family of Training Systems (FTS) documented in the LT2-FTS ICD. The LT2 product line includes all PM TRADE systems that interfaces LT2 systems and supports the U.S. Army's LT2-FTS ICD requirements.

The LT2-FTS is the Army family of interoperable Live training systems based on a Common Training Instrumentation Architecture (CTIA) and a component-based product line that maximizes reusable, common, "plug and play" components and toolsets. The LT2-FTS is the U.S. Army's effort to remove existing Live training systems with redundant requirements, to develop a family of systems that absorbs current capabilities centered on a common architecture, and to expand on those capabilities by eliminating gaps between current and future weapons systems as well as Live U.S. Army and Joint training systems available to support them. The LT2-FTS provide the “Live” domain capabilities for the LVC-IA and interoperate with the “Virtual” and “Constructive” simulation domains to provide a seamless LVC training capability for the soldier (Dumanoir et al., 2006).

2.7.5.1.4 Common Training Instrumentation Architecture (CTIA)

This section provides a description of the Common Training Instrumentation Architecture (CTIA). CTIA is the software framework by which the PM TRADE LT2 strategy will develop product line components that are re-usable and composed to instantiate multiple Instrumentation Training Systems that shall be deployed to Combat Training Centers (CTC), Homestations, and instrumented ranges (PEO-STRI, 2006a). CTIA is the U.S. Army’s product line architecture for the LT2-FTS. For all LT2 products, the LT2 product line objective is to use the CTIA as their main training instrumentation architecture (Dumanoir & Rivera, 2005). The CTIA program provides the protocols, standards and interfaces with other Live, Virtual and Constructive simulation environments. CTIA is also a Future Combat Systems (FCS) complementary program that is a major contributor to the FCS Training Common Components. CTIA represents PEO STRIs common architecture for the Live Training Domain and its strategy to interoperate with other PEO STRI Virtual and Constructive Domains (PEO-STRI, 2006a).

The CTIA is a component-based client-server architecture, which allows for “plug and play” components to interact through the CTIA infrastructure. Figure 29 provides a view of a layered structure of this architecture which includes both wired and wireless communications components, supports several Operating System (OS), and provides the Data Distribution Manager (DDM), CTIA Services, Object Model and Graphical User Interface (GUI) Framework to promote reuse and standardization.

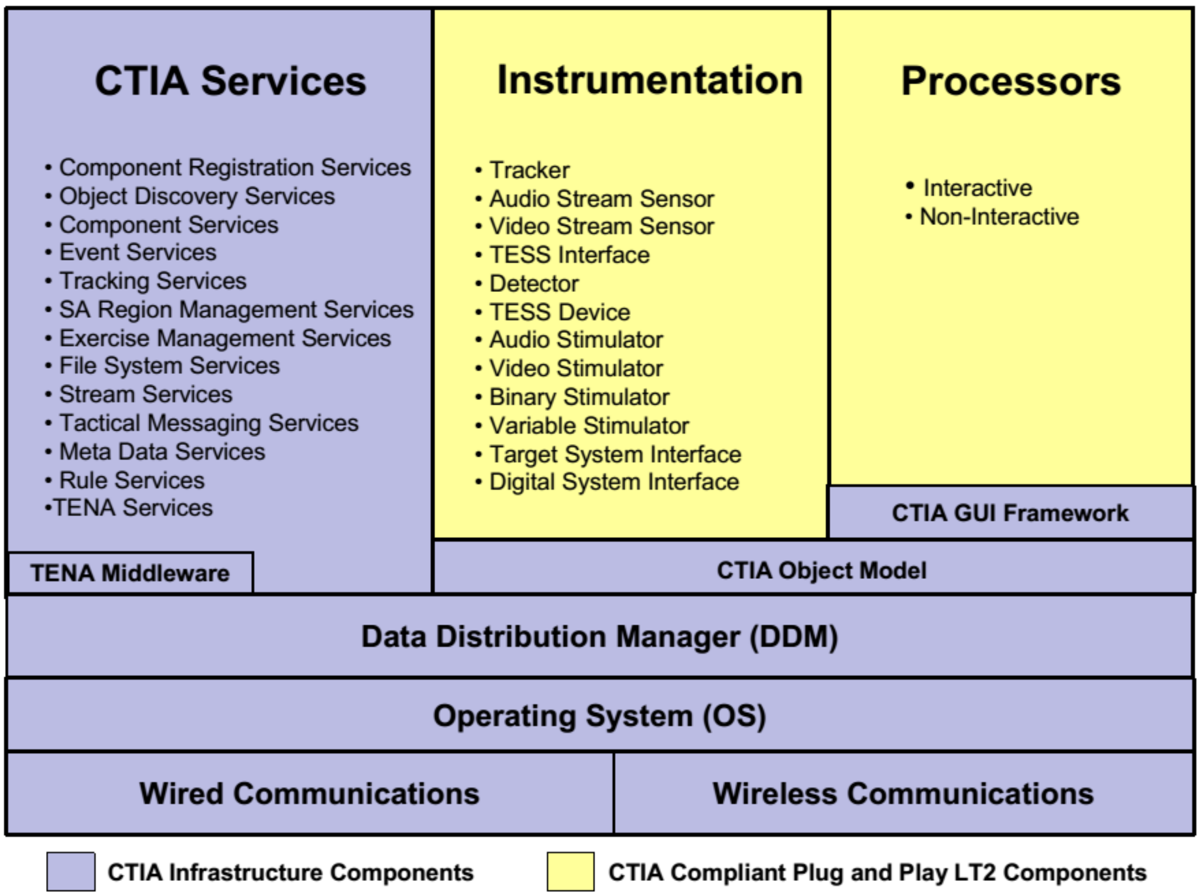


Figure 29: CTIA Layered View

Source: Dumanoir and Rivera (2005)

● CTIA Services – The CTIA Services provide domain-specific services to support plug and play component clients. When deployed, these services will be tailored to account for things such as training exercise scale, available infrastructure, and network variability. The service interfaces use a predefined object data model to ensure component interoperability and remove “stove pipe” systems. These interfaces are defined using the CORBA interface definition language (IDL), which defines object data structures without methods (PEO-STRI, 2006a).

The CTIA Object Models provide methods and higher-level abstractions (e.g. proxies for remote objects). The CTIA services maintain objects that represent exercises, organizations, and participants. It provides services accessible through the Data Distribution Management (DDM) such as unique ID, entity filtering, and brokering control of instrumentation. It provides access to databases for exercise specific and exercise independent data, and encapsulates the databases (PEO-STRI, 2006a).

● Instrumentation – This category of components encapsulates the hardware and software needed to collect data from and control Live entities. Instrumentation is typically associated with Live participants but can be used for simulated. Instrumentation components provide the interfaces to other subsystems and systems such as Tactical Engagement Simulation Systems (TESS), target systems, and Command and Control (C2) systems. In addition, they provide encapsulation of instrumentation such as individual TESS devices, trackers, video cameras, Battlefield Effects Simulators, and control devices in a Mobile Operations on Urban Terrain (MOUT) facility (PEO-STRI, 2006a).

● Processors – This category of components have the capability of producing and consuming all types of CTIA data. This includes tools like After Action Review (AAR) Analysis and Exercise Monitoring as well as Computer Generated Forces (CGF). Processor components can be interactive or non-interactive. Interactive processor components have a user interface and are comprised of the common toolset required across the family of LT2 systems to plan, prepare, execute and evaluate training. Non-interactive processor components include gateways to other simulation or training systems and instrumentation system-based simulations (e.g., Area Weapon Effects). Processors components encapsulate computational functions that have the capability of producing and consuming all types of CTIA data (PEO-STRI, 2006a).

● Communication – These components provide communications between system elements either through wired or wireless networks.

● DDM and Operating System – These components are necessary to complete the definition of the system. DDM provides the back-bone to which other components plug into.

2.7.5.1.5 Relationship between LT2 FTS and other External Systems and Domains

There are several other external systems and architectures that play an important role in enabling the linkages between theLT2 FTS and other external systems and domains. Figure 30 provides a top-level operational view of the external systems and architectures interoperating with LT2-FTS. The LT2-FTS also provides interoperability with other Joint test and training ranges through the TENA as shown in Figure 29 above. The LT2-FTS integrates TENA middleware and a Logical Range Object Model (LROM) with the CTIA services to provide inter-range interoperability within a JNTC training environment.

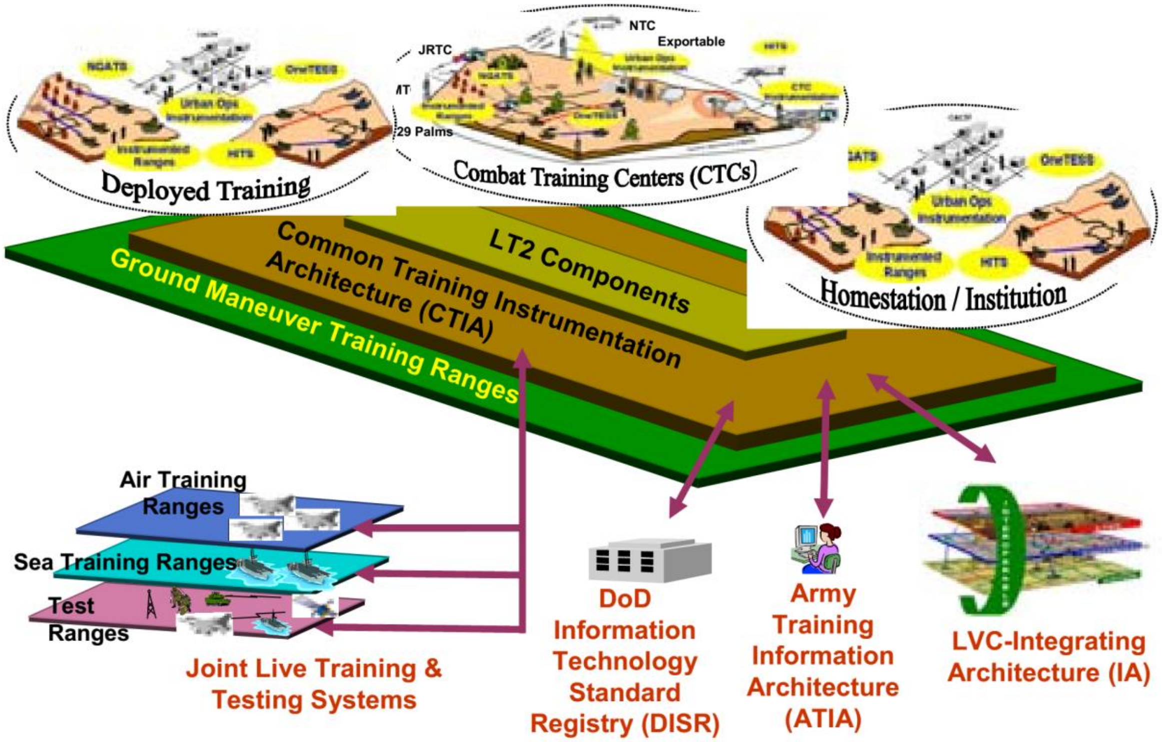


Figure 30: LT2 FTS Operational View

Source: Dumanoir and Rivera (2005)

2.7.5.2 Virtual Component: The Synthetic Environment Core (SE Core)

The Synthetic Environment Core (SE Core) is the U.S. Army's Virtual component of the LVC-IA. The SE Core is the key Virtual program for enabling a common virtual training environment.

The two primary initiatives under the SE Core program are a) the Architecture and Integration (A&I) and b) the Database Virtual Environment Development (DVED) as shown in Figure 31 (PEO-STRI, 2006b). In 2010, the SE Core A&I and the Database Virtual Environment Development (DVED) initiatives were combined into one program – the Common

Virtual Environment Management (CVEM) program. Figure 32 describes the SE core operational view.

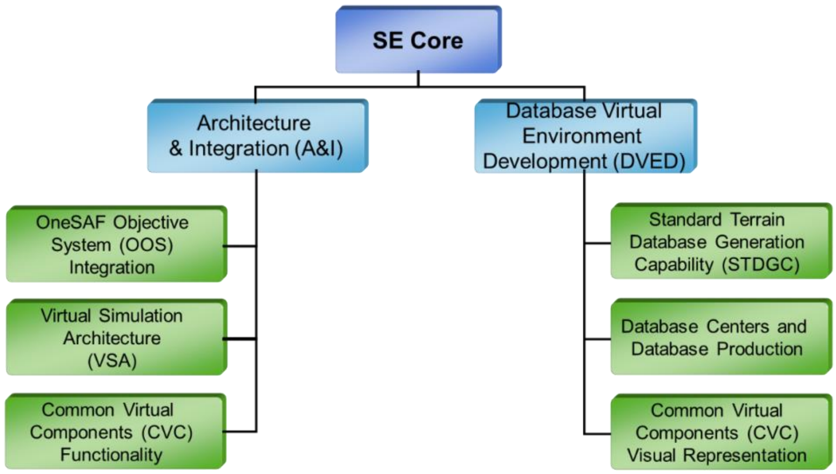


Figure 31: Functional Breakdown of SE Core Program

Source: PEO-STRI (2006b)

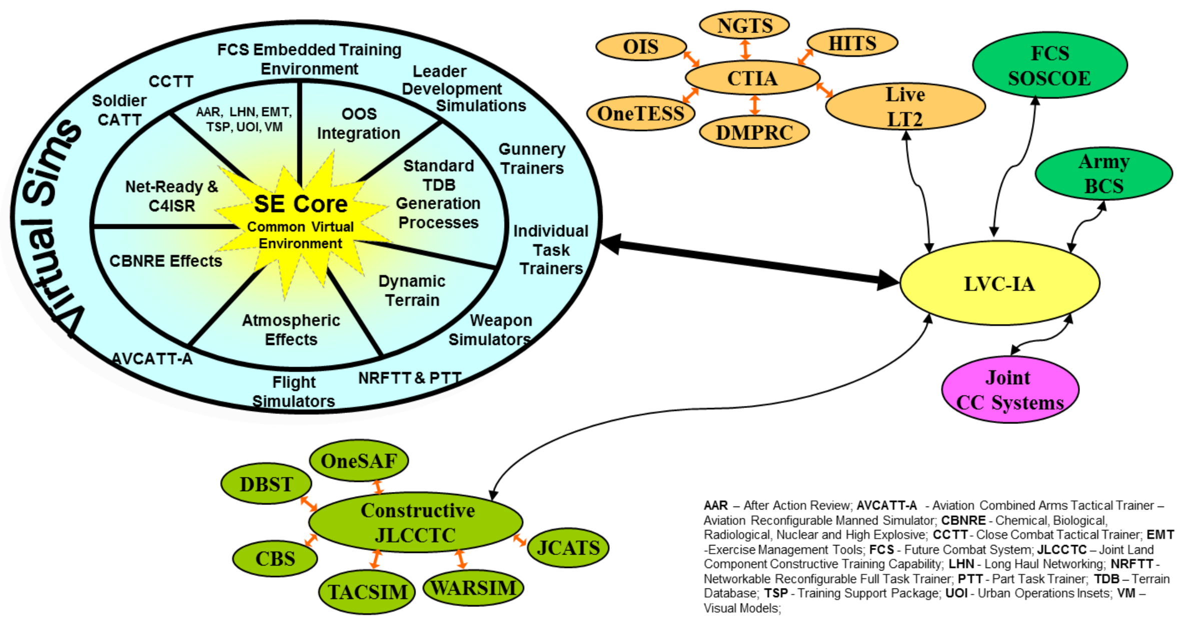


Figure 32: SE Core Operational View within LVC Training

Source: PEO-STRI (2006b)

2.7.5.2.1 Architecture and Integration (A&I)

The Architecture and Integration (A&I)'s main missions is classified (a) OneSAF Objective System (OOS) Integration (b) Virtual Simulation Architecture (VSA) and (c) Common Virtual Components (CVC) Functionality as shown in Figure 31 above.

● OneSAF Objective System (OOS) integration:

A&I is integrating the U.S. Army's One Semi-Automated Forces (OneSAF) into both the Aviation Combined Arms Tactical Trainer (AVCATT) systems and the Close Combat Tactical Trainer (CCTT) (Shufelt Jr, 2006).

● Virtual Simulation Architecture (VSA)

Among A&I's primary mission is for the architecture analysis and development of the Virtual Simulation Architecture (VSA) to provide a Common Virtual Environment (CVE) by developing and integrate existing and new simulation hardware and software products. The CVE, enabled by VSA, will connect Virtual simulation system and non-Virtual simulation systems into a fully integrated and interoperable training capability and will enable soldiers/units training in the Virtual training environment to link with soldiers and units training in the Live and Constructive training environments through the LVC-IA (Shufelt Jr, 2006).

Figure 33 shows the VSA in relationship to other PEO STRI training programs in VSA domain context.

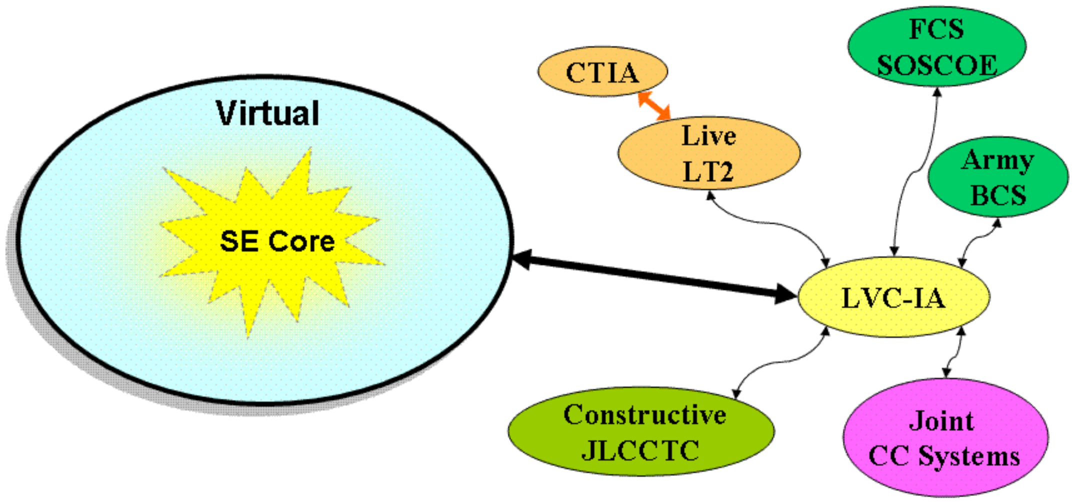


Figure 33: VSA Domain Context

Source: Faulk, Fuchs, Littlejohn, and Kemper

The VSA applies the Product Line Architecture (PLA) concepts to provide a set of reusable products, components, services interfaces, and standards that allow current and future PEO STRI programs to satisfy their service needs (PEO-STRI, 2006a). The VSA is specified in the Product Line Architecture Specification (PLAS) document. The PLAS provides SE Core program stakeholders (end users, clients, customer, developers, etc.) with multiple integrated architectural views of the VSA. The primary focus of this document is product line decomposition, architectural boundaries, and overall interoperability interfaces, which are all necessary for proper component development and use. Figure 34 illustrates the various specifications and architecture views contained in the VSA PLAS (Faulk et al.).

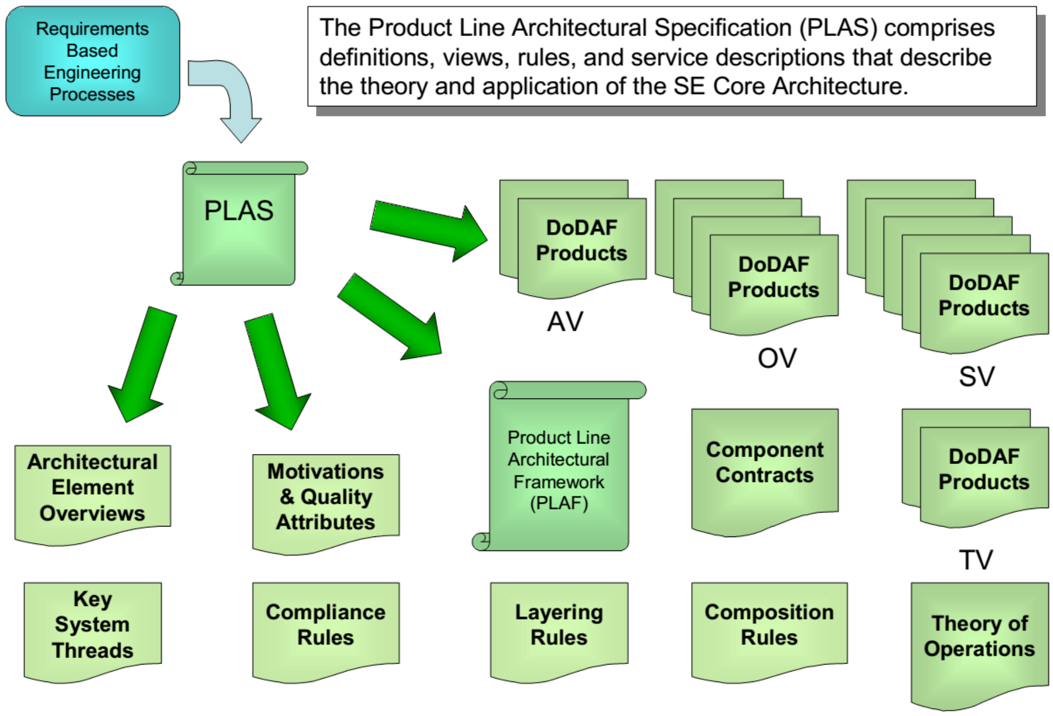


Figure 34: PLAS Document Breakdown

Source: PEO-STRI (2006a)

- Common Virtual Components (CVC) Functionality

The Common Virtual Components (CVCs) enable plug-and-play operation and will be designed to provide common training elements for use within the U.S. Army's Virtual simulation domain. Through commonality, the VSA and CVCs will reduce future development and life-cycle costs.

2.7.5.2.2 Virtual Simulation Architecture Product Line Architecture Framework (VSA PLAF)

The SE Core program is developing the VSA as a common Product Line Architecture (PLA) supporting the development of new and the evolution of current PEO STRI Virtual simulation training systems (PEO-STRI, 2006a).

The VSA utilizes a product line approach that emphasizes systematic reuse and interoperability provides the foundation and guidelines for developing Common Virtual Components (CVCs). One essential view contained within the PLAS is the Product Line Architecture Framework (PLAF). The VSA PLAF view shows the architectural layered organization of the VSA as shown in Figure 35.

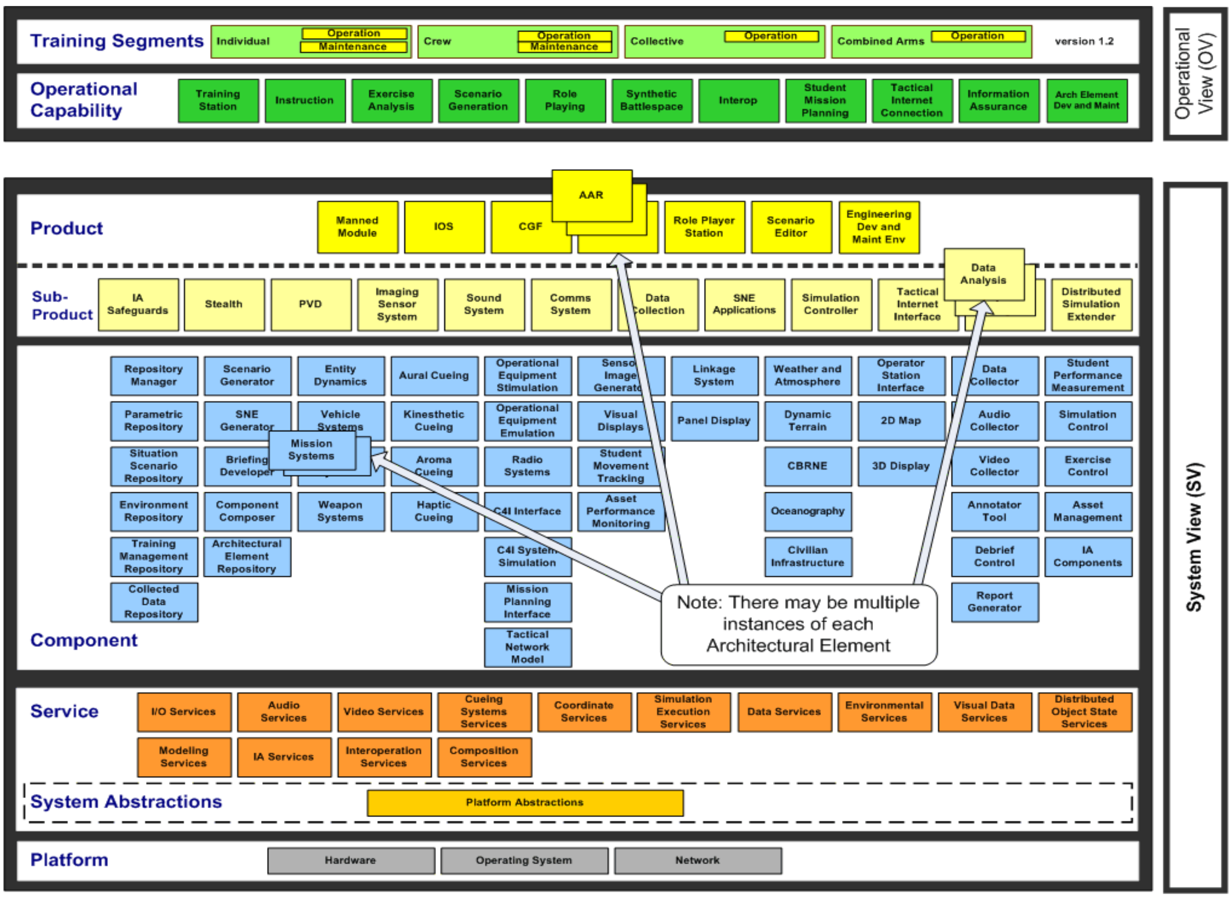


Figure 35: VSA PLAF

Source: PEO-STRI (2006a)

The PLAF is a tool intended to assist the developers of the systems, by helping them identify the architectural components, boundaries, breakdowns, and typical compositional relationships between the layers of the architectural elements (PEO-STRI, 2006a).

The VSA PLAF system view area is divided into the following layers (Faulk et al.):

● Training Segments – Training segments list the major groupings or segments of training systems within the VSA domain.

● Operational Capability – The operational capability layer shows the high-level training operational activities performed by the domain training systems. The operational activities describe the major tasks/functions that are required for the domain training system sites to accomplish their missions.

● Product – Products are stand-alone, end-user visible functionality representing the very high level applications or application suites that are typically deployed as a unit. They represent significant architectural pieces of a training system, such as an after action review (AAR) or instructor operator station (IOS). The VSA defines the specific interface protocols to facilitate the Product level interoperability.

● Subproduct – Subproducts are just smaller scale products and maintain the same characterizations as a product. The hardware analogy is that of a line replaceable unit (LRU), allowing substantial subsystem level functionality to be swapped out within a training system. Subproducts will often be deployed into a training system as a collection composing a full product, however, they may be deployed individually as necessary to meet a specific training system's needs. For example, a simulation controller Subproduct may be deployed at an after action review workstation allowing an operator to perform exercise control from that physical station area.

● Component – Components are the systematically reusable building blocks of Products and Subproducts. This is where the majority of the software is produced within the VSA framework. Components are built on the VSA services providing further software reuse, portability, and interoperability.

● Service – The VSA services are a set of common software service interfaces that provide the framework or infrastructure on which VSA common components are built. The common services promote systematic reuse and consistency for component distribution, component and service discovery, data models, data distribution, component communications/messaging, scaling, and portability across the VSA common components.

● Platform – The platform layer represents the host hardware, operating systems, and network technology supported by the VSA. This is typically commercial off-the-shelf (COTS) or open source and is not being developed by SE Core A&I. However, the VSA will, specify requirements on this layer such as real time execution support.

2.7.5.2.3 Database Virtual Environment Development (DVED)

Database Virtual Environment Development (DVED)'s primary mission is to generate *correlated simulation system runtime databases* rapidly for supported simulation systems. A master SE Core database is populated from a union of multiple authoritative data sources by using a DVED-defined software architecture, processes and a suite of commercial and government-off-the-shelf (GOTS) database development software tools. The DVED architecture and tools will enable the generation of master SE Core databases in hours or days versus months. The DVED effort also develops common virtual vehicle models, common virtual sensor simulation software and virtual simulation components. With SE Core as the foundation, the U.S. Army will leverage existing Virtual simulation systems as well as expand the overall use of Virtual simulation systems within Live, Virtual and Constructive environments to support ongoing U.S. Army transformation (PEO-STRI, 2012).

2.7.5.2.4 SE Core Standard/Rapid Terrain Generation Capability (STDGC)

● Overview of STDGC - The SE Core Standard/Rapid Terrain Generation Capability (STDGC) is intended to create a single unified process that supports the generation of all of the Virtual and Constructive databases required by confederate simulation systems (PEO-STRI, 2006a). The STDGC has two major functionality components;

The first is the generation of a single unified Master Database (MDB) that is built at the highest level of data resolution possible from available government and commercial sources. The MDB is constantly be updated as new data sources are acquired and as the geo-political climate changes.

The second functionality piece is that of a database tailoring and formatting tool that tailors the MDB to the training objectives, systems capabilities, and run-times formats required by the confederate training systems (PEO-STRI, 2006a).

● Goal of STDGC - The STDGC has the requirement to generate databases that are 180 km x 180 km in size with a data resolution equivalent to National Geospatial-Intelligence Agency (NGA) DTED level 3 (terrain surface resolution) and an urban inset within that database that is 2.5 km x 2.5 km with an equivalent resolution of NGA DTED level 5 (terrain surface resolution) to support MOUT/Urban operations. The MDB must be produced within 96 hours using COTS tools, open formats, and automated processes.

● Implementation concept of the STDGC - The implementation concept of the STDGC is shown in Figure 36 and 37.

For the first part of the implementation the initial concept is to use COTS to generate the MDB. Conceptually, the MDB consists of multiple open formats that facilitate a layered approach to the accessing and storage of the MDB. The MDB is designed to accommodate data for the entire world but realistically it only contains data for those parts of the world deemed important (e.g., home stations, training areas, areas of current and potential future military operations, other areas of interest). The MDB must also be maintained at the highest data resolution available from government and commercial sources and must also support current environmental data models (e.g., the OOS EDM).

The second part of the implementation involves the generation of the individual databases required for the confederate systems. For example, the AVCATT system would involve the generation of the visual and sensor databases in the L3 format, the OOS Semi-Automated Forces (SAF) databases and maps (electronic and paper). To achieve this, the conceptual implementation of the Real-Time Database Generation Toolkit (RDGT) would be to run off-line to create static databases in each of the required formats. The RDGT will have three major tasks: extraction of the data required for the training mission from the MDB, thinning, integration, and manipulation of the data to the training and system requirements, and finally formatting the data to the required format for the respective software application.

- The first task, extraction of the data from the MDB, will be through a government-owned API to facilitate the reuse and interchangeability of the data thinning, integration, and manipulation subroutines within the RDGT.
- The second task of thinning, integration, and manipulation will be controlled by a scripted process that resolves capability differences between differing simulation systems and provides correlated data to each simulation system in the confederation.
- The third task is the formatting of the correlated data to the individual simulation systems.

To this end, the government will develop and maintain an API for writing data to simulation systems. Individual system vendors will be responsible for developing software plug-ins that conform to this API and will write the data into their individual database formats. These plug-ins will ensure the preservation of the data correlation and accuracy requirements and that the data is formatted and structured to work with their individual systems.

Other aspects of the STDGC concept include the automatic testing of the integrity of the MDB, distributed production facilities that provide local interaction with area commands in the generation of areas of the world, and alignment of the STDGC with other data initiatives within the military (ex. RD3, J-GES, PDI). Also, STDGC will support the generation of databases for FCS.

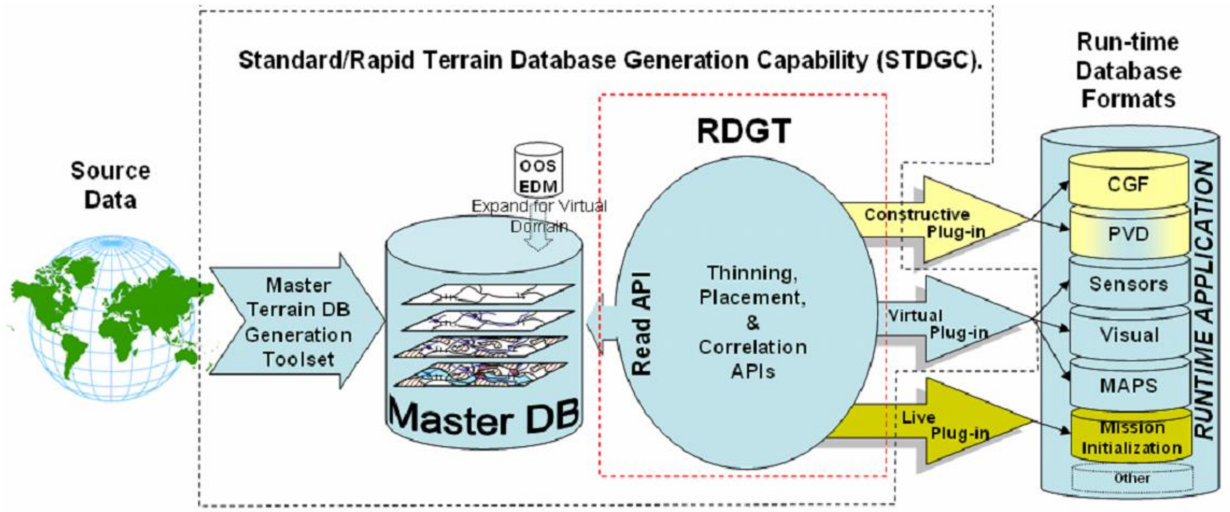


Figure 36: Overall STDGC Process Concept

Source: PEO-STRI (2006a)

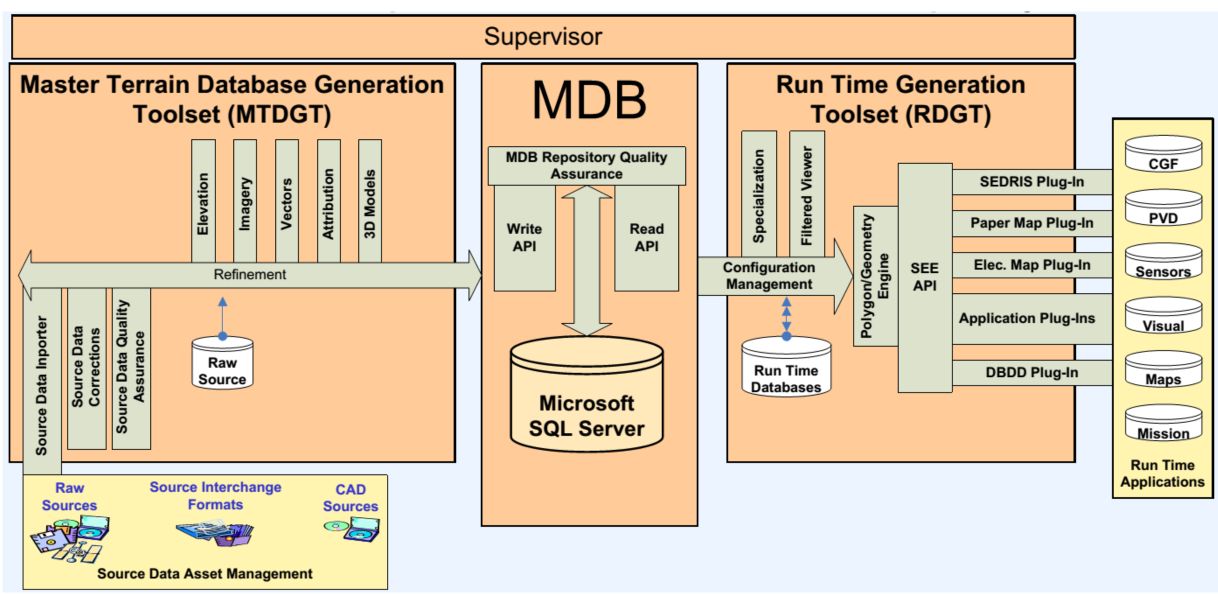


Figure 37: Detailed STDGC Process

2.7.5.3 Constructive Component: Joint Land Component Constructive Training Capability (JLCCTC)

The constructive simulations being regularly used for U.S. Army training are part of what was called the Joint Land Component Constructive Training Capability (JLCCTC). The goal of JLCCTC is to provide a federation of eight models that can interoperate in the short term, while migrating over the long term to an objective system with fewer simulations that are more highly integrated and use less communications bandwidth (Shanley, 2007).

JLCCTC is a modeling and simulation software capability that contributes to the joint training functional concept and the U.S. Army training mission area by providing the appropriate levels of modeling and simulation resolution as well as the fidelity needed to support both U.S. Army and joint training requirements. JLCCTC is composed of two separate federations, JLCCTC Multi-Resolution Federation (MRF) and JLCCTC-Entity Resolution Federation (ERF). Figure 38 provides an overview of the JLCCTC architectures (PEO-STRI, 2006a).

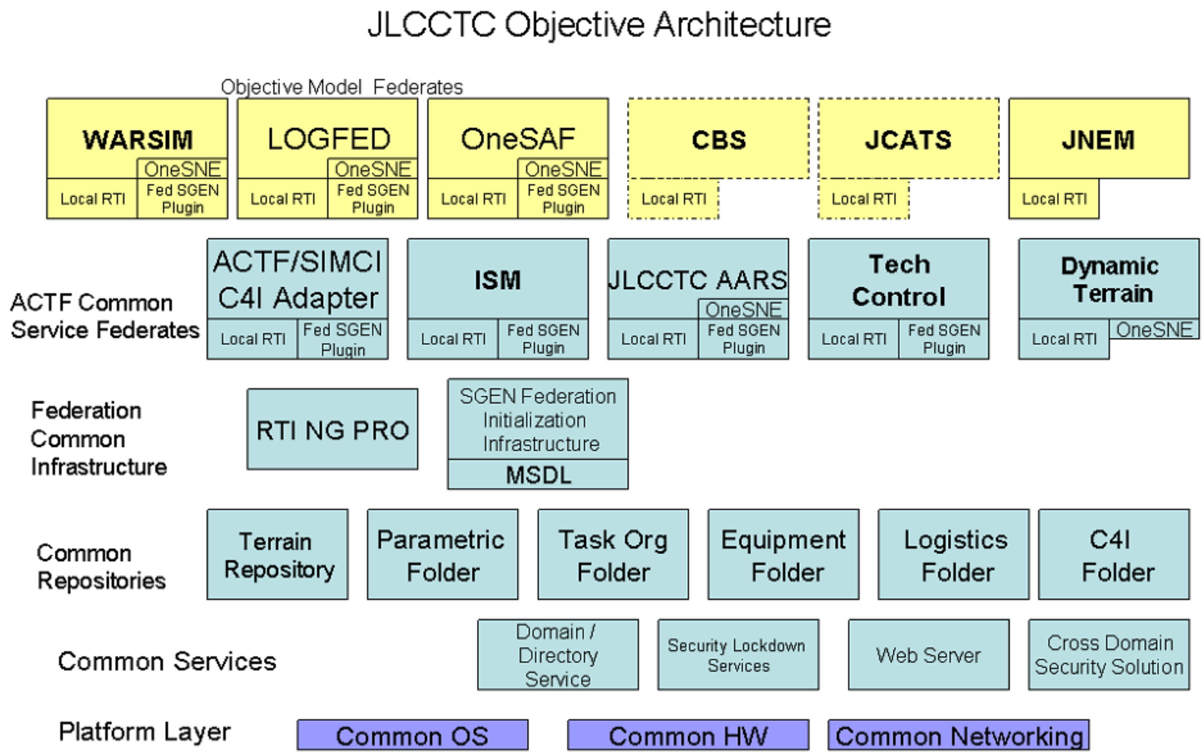


Figure 38: JLCCTC Objective Architecture

Source: PEO-STRI (2006a)

2.7.5.3.1 JLCCTC Multi-Resolution Federation (MRF)

The Multi-Resolution Federation (MRF) is a federated set of constructive simulation software that is supported by commercial software and commercial-off-the-shelf hardware that will support training of commanders and their staffs in maneuver, logistics, intelligence, air defense and artillery. The JLCCTC MRF FOM is maintained and Configuration Managed for PEO STRI by the MITRE Corporation is shown in Figure 39. The federate models are connected by a combination of the standard high-level architecture run-time infrastructure, distributed interactive simulation, custom interfaces, the master interface and point-to-point (PEO-STRI, 2006a).

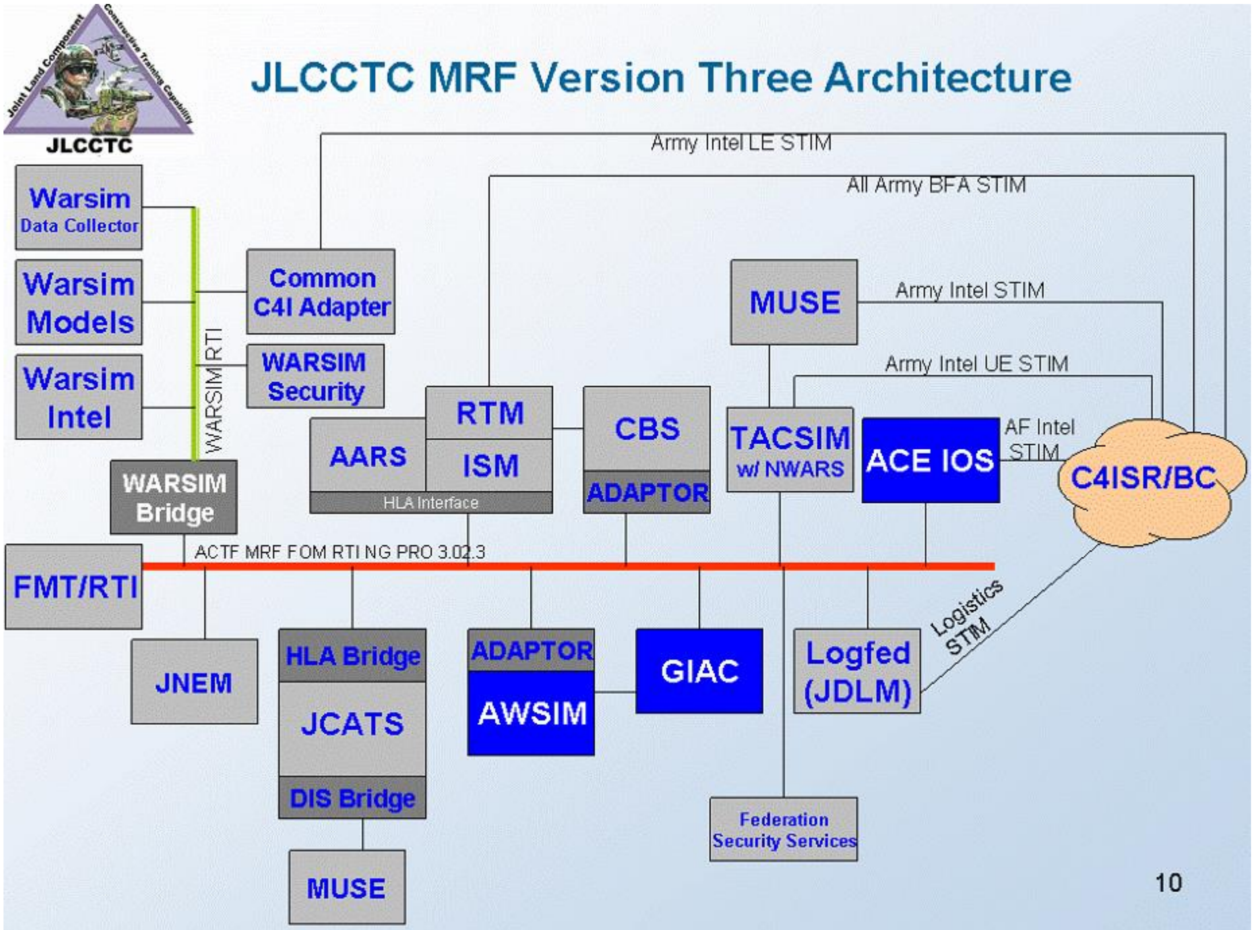


Figure 39: JLCCTC MRF V3 Architecture

Source: PEO-STRI (2006a)

JLCCTC provides the simulated operational environment in which computer-generated forces stimulate and respond to the Mission Command (MC) processes of the commanders and staffs. JLCCTC models will provide full training functionality for leader and battle staff for the Army and the joint, intergovernmental, interagency and multinational (JIIM) spectrum. JLCCTC provides an interface to MC Systems allowing commanders and their staffs to train with their organizational real-world MC equipment.

2.7.5.3.2 JLCCTC MRF-Warfighters' Simulation (WARSIM)

The JLCCTC MRF-WARSIM trains Army commanders and their staff in support of Command Post Exercises (CPXs), Warfighter Exercises (WFXs), and Mission Rehearsal Exercise (MRXs). WARSIM is a next-generation, large-scale constructive wargaming system, developed for U.S. Army command and control training. It is being developed to replace the current legacy simulation systems, e.g., Corps Battle Simulation (CBS) and Tactical Simulation (TACSIM). WARSIM is a significant advance in modeling and simulation technology deploying a wide range of resolution, fidelity and abstraction, depending on its specific use. WARSIM is a distributed, constructive wargaming simulation, designed to create a single, seamlessly integrated synthetic battlespace, including a common environmental and operational picture. Interfacing with C4I functions and equipment in the field to provide the interface between the synthetic battlespace and the training audience, WARSIM creates a training environment intended to be indistinguishable from the real-world by the training audience.

WARSIM is a constructive simulation system used to train commanders and staffs at brigade, division, corps and echelons above corps. When conducting an exercise, it can be viewed as three layers. At the top is the training audience. The training audience consists of the commanders and staff of the units to be trained, organized and equipped as they would be in an operational setting. Their command posts may be field locations or they may be at a training center, but they are equipped with the tactical C4I devices that would be used to conduct actual operations.

The second layer is a set of "role players." These are people who perform the roles of the subordinate commanders and staff of the training audience. They interact with the training audience via tactical communications and C4I tactical messages to provide the stimuli that allow a training exercise to proceed. The role players also control the third layer of WARSIM, which is the computer simulation of the battlespace. The role players provide the military skills to direct the simulated units and to represent the persons with which the training audience expects to interact. In particular, the role players provide the person-to-person voice interactions that characterize Army command and control even in this digital era. At this point, there is some ability to exchange message traffic between the simulated units and the training audience without role player intervention, but this accounts for only a small part of the interaction. The three-layer structure is shown in Figure 40 (PEO-STRI, 2006a).

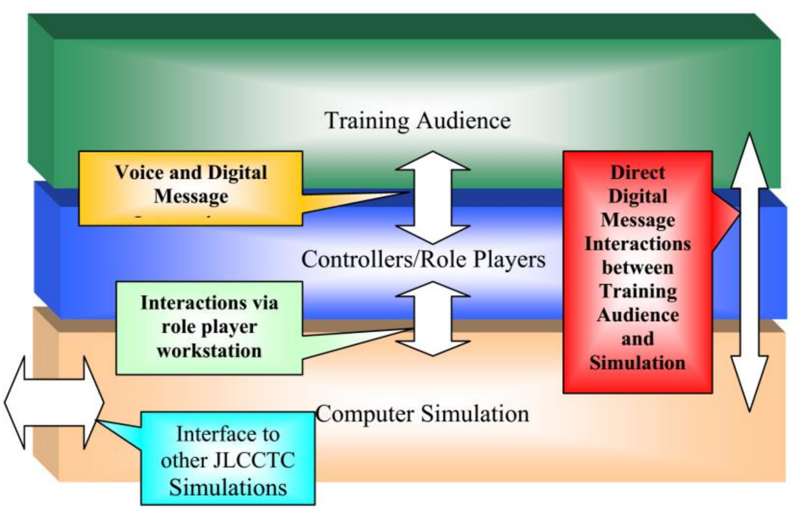


Figure 40: WARSIM 3-Layer Architecture

Source: PEO-STRI (2006a)

Since the training audience operates with its own equipment during an exercise, the boundary of WARSIM consists of the lower two layers and the interfaces to the training audience. The simulation component of WARSIM is a real-time model of military forces on a highly detailed representation of the terrain. It provides automated units at company level that are capable of accepting orders from role players, planning the execution of those orders and controlling the actions of subordinates (e.g., platoons). The simulation provides a level of resolution such that positions of individual vehicles can be determined. Resolution of combat engagements occurs via simulation of the weapons effects as affected by both the terrain and the ability and condition of the simulated units. This level of detail allows the simulation to provide detailed output to role players and to the training audience.

The System Architecture is a composition of the WARSIM hardware and software along with COTS and Government-Off-The-Shelf (GOTS) software products. Communication between elements of the system is accomplished by use of the WARSIM Federation Object Model (FOM) and the HLA Run Time Infrastructure (RTI). Figure 41 illustrates the abstract relationship between the major components. The Computer Simulation piece can be viewed as four separate partitions:

- Interface to the Training Audience
- Simulation
- Controller Interface
- Infrastructure

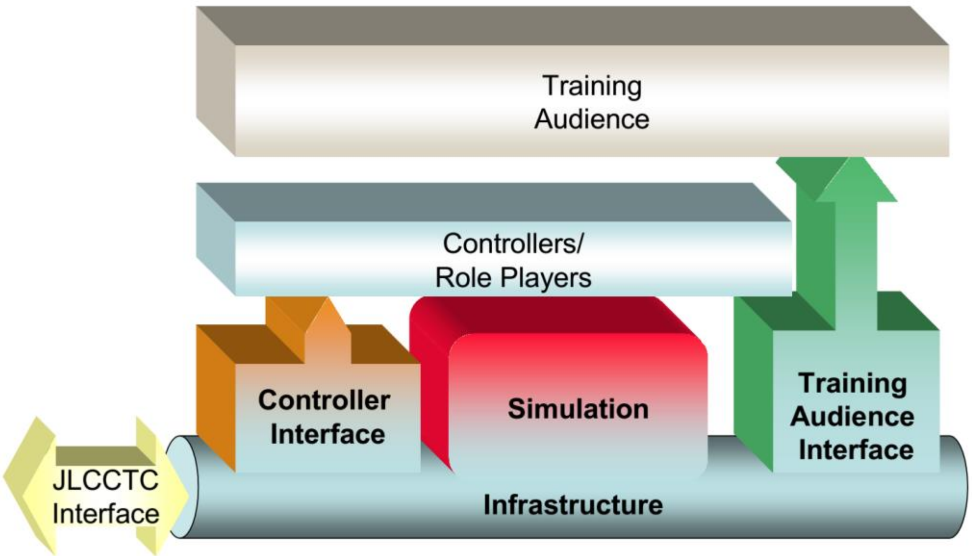


Figure 41: WARSIM Abstract System Architecture

Source: PEO-STRI (2006a)

The three layer structure discussed earlier and shown in Figure 40 can be seen in Figure 41. The lowest layer represents the hardware and software that is installed at a training center. It is divided into four partitions.

● The simulation partition models the battlespace and battlespace elements that model the combat activity used to stimulate the training audience.

● The training audience interface partition connects the training audience C4I equipment or surrogates with the simulation and with the controller stations.

● The controller interface partition allows the simulation controllers and analysts to interact with the training audience, control simulated units, and monitor the simulation system.

The infrastructure partition provides common services required by all components of the simulation system (PEO-STRI, 2006a).

2.7.5.3.3 JLCCTC-Entity Resolution Federation (ERF)

JLCCTC ERF is a federation of simulations, data collection and after-action review tools as shown in Figure 42. The JLCCTC ERF FOM is maintained and Configuration Managed for PEO STRI by the MITRE Corporation. It stimulates the Mission Command Networks and Systems to facilitate battle staff collective training by requiring staff reaction to incoming digital information while executing the commander tactical plan. The targeted training audience is comprised of brigade and battalion battle staffs, functional Command Post (CP) training and full CP training (PEO-STRI, 2006a).

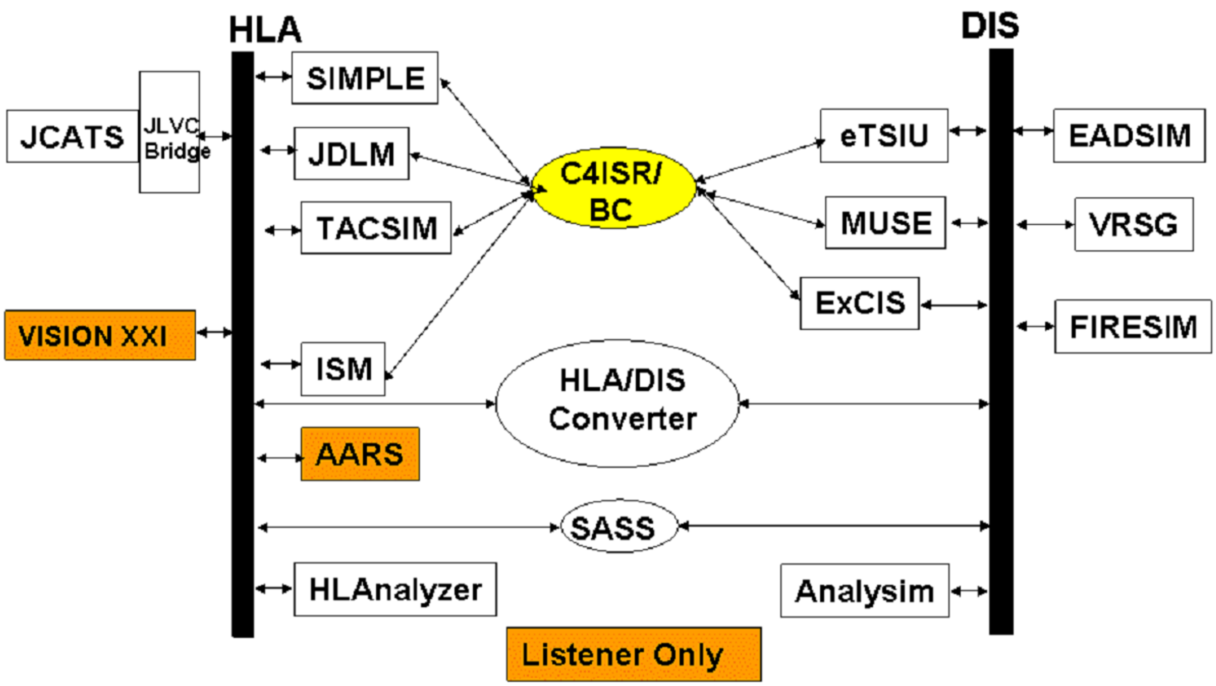


Figure 42: JLCCTC ERF V3 Logical Block Diagram

Source: PEO-STRI (2006a)

2.7.5.3.4 JLCCTC ERF- One Semi-Automated Forces (OneSAF)

OneSAF is a composable CGF that represents a full range of operations, systems, and control processes from the individual combatant and platform level.

The PLAF is a mechanism to organize, categorize, and define the layered software structure to incrementally meet the OneSAF requirements. The PLAF identifies functionally relevant software components that can be used as building blocks for higher level functionality.

Within the Product Line Architecture Specification (PLAS), the PLAF provides a static view of the System Compositions, Products, and Components that comprise the OneSAF Architecture. See Figure 43. The OneSAF Architectural approach facilitates meeting both current and future undefined requirements (PEO-STRI, 2006a).

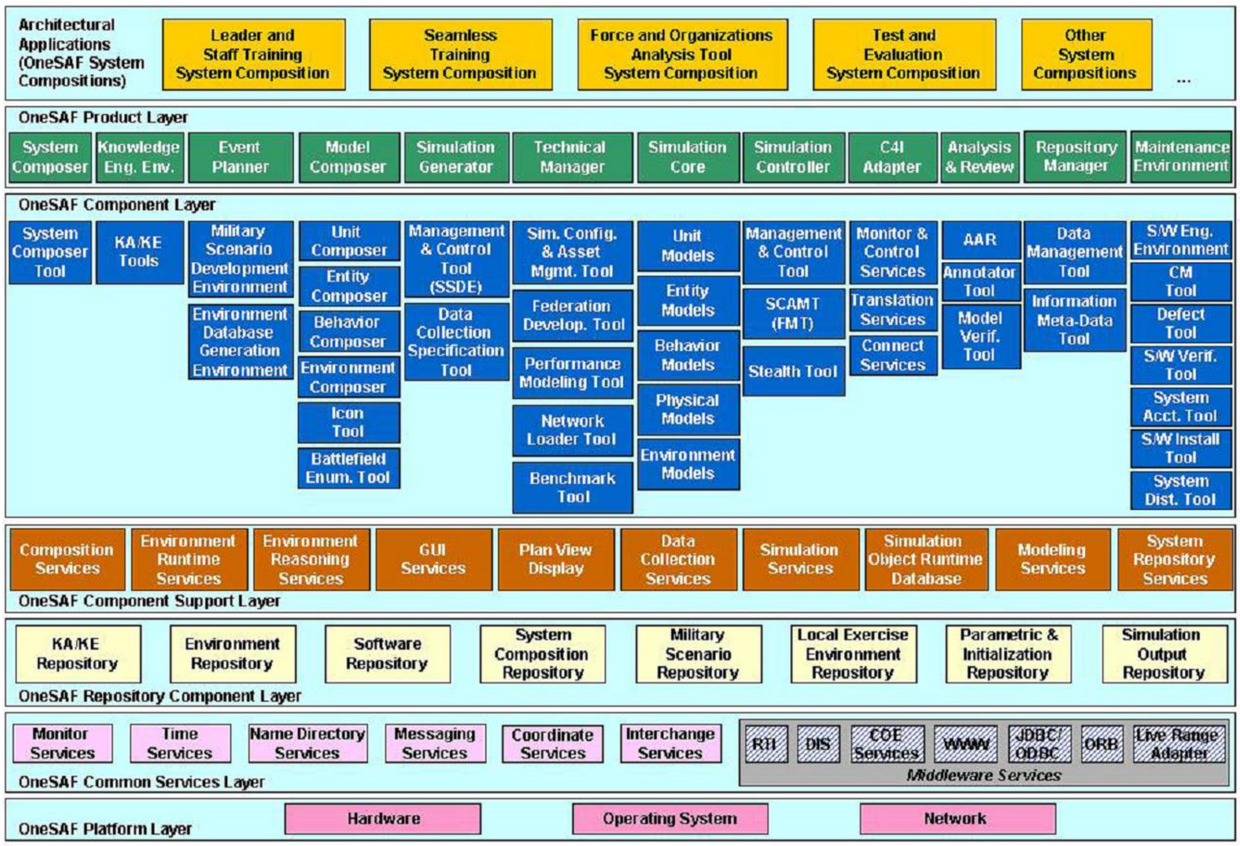


Figure 43: OneSAF Product Line Architecture Framework

Source: (Logsdon & Wittman, 2007)

2.7.6 Goal of the U.S. Army LVC-IA

According to Dumanoir et al. (2006); Dumanoir et al. (2004), the goal of the U.S. Army LVC-IA is to seamlessly interconnect and ensure interoperability with Joint National Training Capability (JNTC), Joint Land Component Constructive Training Capability (JLCCTC) Army Constructive Training Federation (ACTF), Army Training Information Architecture –Migrated (ATIA-M), CTIA, and SE Core.

2.7.7 Section Summary

Currently, there are many challenges in integrating Live, Virtual, and Constructive simulation-based training events: a) difficulties in integrating legacy simulations, b) difficulties in integrating different types of simulations, c) extensive scheduling, preparation, and support are needed to execute effective integrated events, and d) large areas of uncertainty exists regarding the technical aspects of achieving interoperability between legacy and newly developed simulations, or components of simulations (e.g., SAF) (Shanley, 2007).

If the above challenges are solved and architecture integration initiatives are achieved to the degree, LVC-IA will have some positive effects on the quality of training by the 2016 timeframe. Eventually, the LVC-IA will increase training effectiveness and efficiency by expanding the battle space for training and minimizing cost by standardizing hardware, software, and infrastructure between live, virtual, and constructive simulations, simulators, and instrumentation.

M&S communities expect that U.S. Army programs such as the LVC-IA, along with the state-of-the-art science and technology, will greatly increase the capabilities and interoperability of the LVC simulation, resulting in a more accurate replication of the real environment.

2.8 Common Standards, Products, Architectures and/or Repositories (CSPAR)

The PEO STRI Policy on the Use of Common Standards, Products, Architectures and/or Repositories (CSPAR) defines policy for the designation and use of common products and the identification of communication and interface standards, data models and architectures which facilitate and ultimately reduce the cost of the integration and interoperability of Live, Virtual and Constructive (LVC) capabilities across PEO STRI. This reference document was established by a committee comprised of Chief Engineers from each of the PEO Project Mangers. It includes a reference set of recommended standards, protocols, components, architectural approaches and data repositories (Logsdon & Wittman, 2007).

2.9 Research Gap

In Chapter 2, I have discussed several important topics to improve the interoperability, integration, composability and reuse of the LVC simulation. The following sections describe identified gaps in developing a seamless LVC simulation environment.

2.9.1 Complex Integration

To integrate a Virtual or Constructive simulation system into a LVC simulation, it may be necessary to upgrade several existing applications. The more applications that are integrated, the more complex it becomes to integrate an additional application. Further, when upgrading an application, existing functionality may be affected, requiring even more work. This complexity makes it hard to adapt to new SSA (Gustavsson et al., 2009). Therefore, cutting-edge technologies, tools, and simulation architecture frameworks are needed to reduce the complexity of developing simulation applications in the emerging LVC simulation.

2.9.2 Long Time-to-LVC User-Usage

For a higher level of interoperability between LVC simulation systems, one possible solution is either to develop a new single future LVC SSA or to use of bridge such as gateway and middleware for LVC simulation as shown Figure 44. However, by this time, no new LVC SSA has been developed as planned and framework/gateway/middleware has been used for LVC simulation.

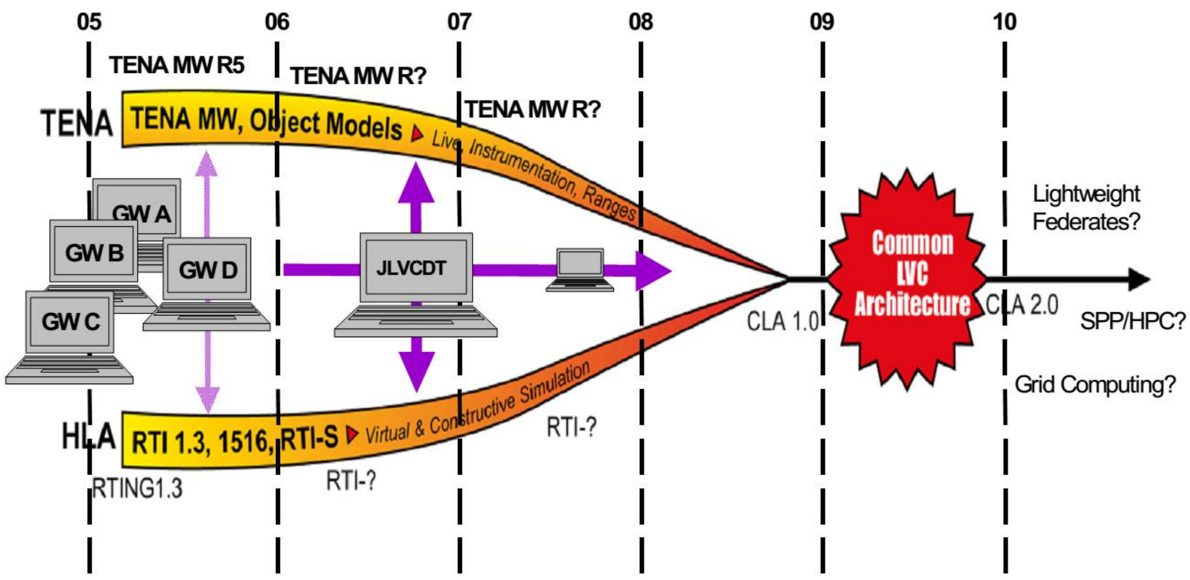


Figure 44: Common LVC Architecture Vision

Source: W. Bizub et al. (2006)

M&S community might expect that the U.S. Army LVC-IA and DoD LVCAR programs will remove many of these shortfalls regarding LVC interoperability, leading to a training environment that more closely replicates the combat environment. However, the time to-LVC user-usage is how long it takes to develop a new function by integrating a number of applications that together satisfies a new need or meet sudden requirements. Since such an integration is complex, the time-to-market will be long.

2.9.3 High Cost

Each additional application that is integrated potentially will cause more integration process than the last one. This makes integration costs increase rapidly. The more applications that are integrated, the more complex it becomes to integrate an additional application. Further, since each additional application that is integrated may affect several other existing applications, life-cycle costs will also remain high.

2.9.4 Inflexible Integration

The technical issues that needed to be resolved were unique to particular events. To change the way a number of applications are integrated may require re-integration of the applications all over again because of the interdependency between the applications. Integration is rigid, and inflexible.

CHAPTER THREE: METHODOLOGY

This chapter describes the detailed research methodology. As mentioned in Chapter 1, the ultimate purpose of this research is to enhance the interoperability, integration, composability and reuse in LVC simulation environment. To achieve the purpose, the goal of this research is to provide an *agile roadmap for the Live Virtual Constructive-Integrating Training Architecture (LVC-ITA).*

The methodology for an agile roadmap of the LVC-ITA provides a complete step by step process for examining pertinent issues and provides solutions to resolve problems. The research methodology follows as shown in Figure 45.

3.1 Flow Chart of Methodology

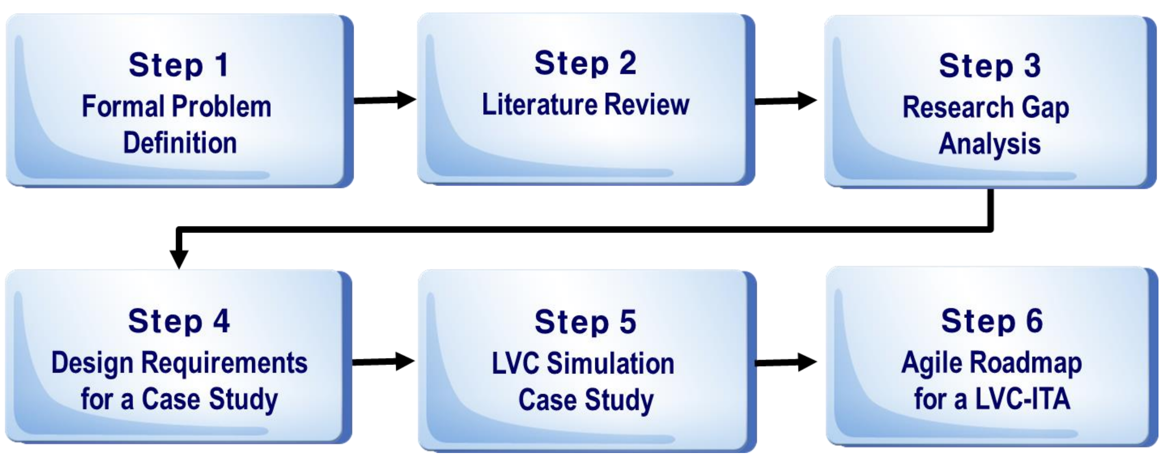


Figure 45: Flow Chart of Methodology

3.2 Description of Methodology

The methodology is six steps in total. In the following sections, I describe in detail, each of the steps as shown in Figure 45 above.

3.2.1 Step 1: Formal Problem Definition

In this section, we describe the formal problem definition. A major problem with LVC simulation environment is that a seamless LVC simulation is limited. The primary objective of Step 1 is to develop a clear understanding of the problems to be addressed in the current M&S environment. The identified problems were described in detail in Section 1.2. The problems are as follows:

- Problem 1: Inherent Limited Interoperability between the Different SSAs.
- Problem 2: Many Issues in Integrating LVC Assets.
- Problem 3: Decentralized Management and Development of SSAs and LVC Assets

Due to these problems, we need to study prior and/or current approaches for seamless LVC simulation.

3.2.2 Step 2: Literature Review

The methodology begins with a thorough literature review. A large amount of relevant literature has been collected. The state-of-the-art technology and skill with respect to interoperability, composability and integration were investigated. The literature review provided a sufficient basis to identify the current state, the functional requirements, the priority and the capabilities for LVC interoperation. If there are gaps, additional literature review was often conducted. The literature review was a continual process rather than a single step taken to achieve the purpose of research.

3.2.3 Step 3: Research Gap Analysis

● Step 3-1: Comparative analysis for multiple SSAs – analyzing prior works related to types, organizations, development and evolution processes for different SSAs. The objective of Step 3-1 is to understand the differences and technical incompatibilities of the SSAs.

● Step 3-2: Analysis of capabilities and limitations for various SSAs – identifying capabilities and limitations on the currently used SSAs.

● Step 3-3: Analysis and evaluation of previous methodologies and procedures– identifying limitations and shortfalls from related research.

● Step 3-4: Defining needs and requirements for an agile LVC-ITA – identifying research gaps and functional requirements for supporting the LVC interoperability. The identified research gaps are as follows: (a) Complex Integration, (b) Long time to LVC user-usage, (c) High cost and (d) Inflexible integration.

3.2.4 Step 4: Design Requirements for a Case Study

In Step 4, a set of detailed requirements was derived from M&S user communities. A successful roadmap must address and solve all the major issues related to making the development and widespread use. In considering the design of an agile roadmap for the LVC-

ITA, we kept four important design requirements for the LVC simulation case study in mind. I wanted an approach that:

- meets the needs of highly interactive real-time applications.
- should be sufficiently flexible to support interoperability regardless of the SSAs being used in the simulation environment (or federation) (e.g., DIS, HLA 1.3, HLA 1516, HLA evolved, TENA, CTIA, etc.), without requiring changes to the existing native simulation systems (or federates).
- has simple/flexible connection and integration.
- takes short time for LVC users.

3.2.5 Step 5: LVC Simulation Case Study

The detail descriptions of the case study for LVC simulation are explained in Section 4.0.

3.2.5.1 Step 5-1: Designing an LVC Simulation Case Study

Step 5-1 presents the components that consist of LVC simulation case study. Based on the results of Step 1, 2, 3 and 4 above, the objective of Sub-step 5.1 is to enable the selection of alternatives. First, we planned a scenario for LVC simulation case study. Second, we identified the viable alternatives to execute the scenario. Third, we provisionally examined these alternatives by the design requirements of Step 4 that were used for evaluation and eliminated the obvious duds. Fourth, we selected the remaining candidates for further consideration. Fifth, we analyzed the alternative solutions. Lastly, after exchanging and sharing knowledge with researchers at SIL in UCF, we selected the final alternative for the LVC simulation case study (minimal simulation environment instantiations).

The identified alternatives as a component of the LVC simulation case study were a) AddSIM, b) SIMbox, c) VR-Forces, d) Data Logger, and e) WebLVC. This case study reflects current LVC simulation's technologies. A brief description of each component follows. The detailed descriptions on these components of the case study appear in Section 4.4.

3.2.5.1.1 Scenario Concept of the LVC Simulation Case Study

We planned the scenario for the LVC simulation case study. The scenario is an Air Defense Engagement as shown in Figure 46.

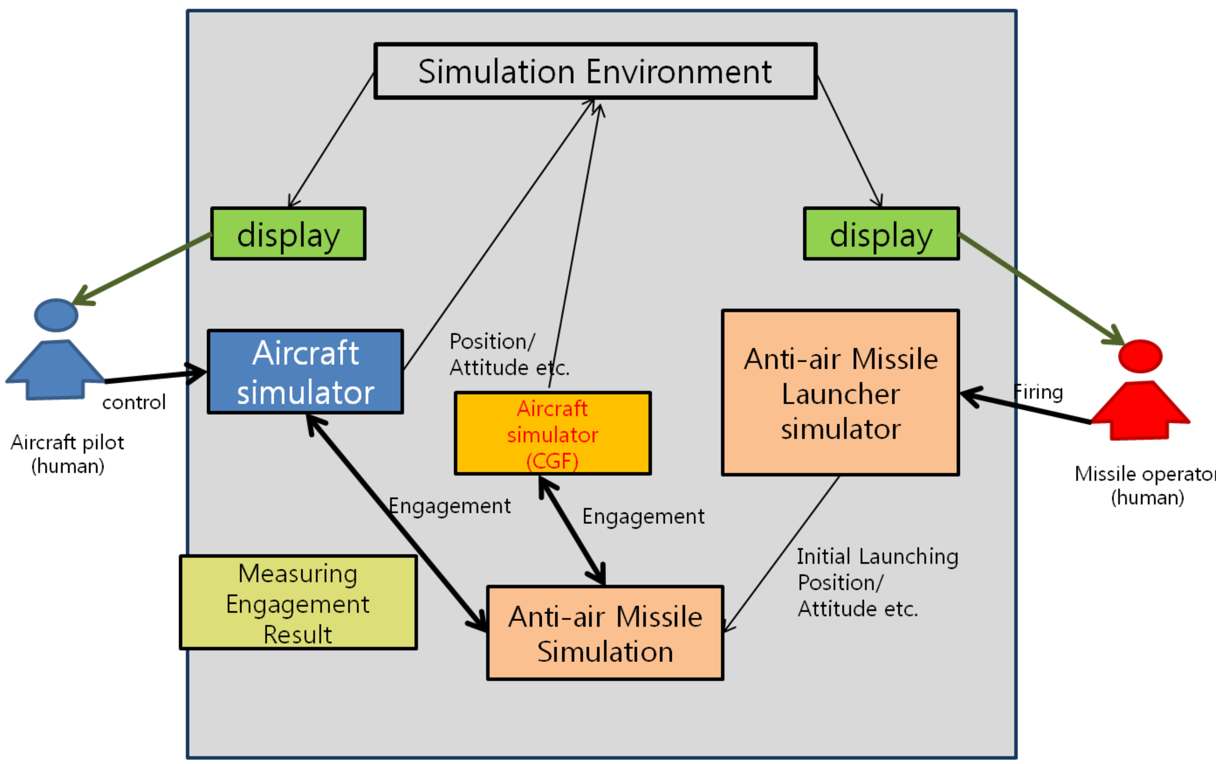


Figure 46: Scenario Concept of LVC Simulation Case Study

The target simulation systems for this scenario as are follows;

● Virtual Flight Simulator.

● Virtual Surface to Air Missile (SAM) Simulator.

● Constructive Simulation System for Computer Generative Force (CGF).

● Engineering Level Model for measuring engagement result.

3.2.5.1.2 Component Based Integrating Simulation Environment (AddSIM)

AddSIM is a component-based weapon system simulation environment using engineering models of weapon systems. The first version of AddSIM was developed through a core technology R&D project of the Agency of Defense Development (ADD) with SimNet, South Korea from 2009 to 2011 (Lee, Lee, Kim, & Baik, 2012). The main goal of AddSIM is to enhance interoperability, reusability, and composability of weapon simulation models (Kim, Oh, & Hwang, 2013).

3.2.5.1.3 SIMbox

The SimiGon has developed a simulation system of Flight and Surface-to-Air-Missile (SAM) in SIMbox simulation platform that is a Commercial off the Shelf (COTS) simulation system for training.

3.2.5.1.4 MÄ K VR-Forces

We choose MÄ K VR-Forces as a Constructive simulation system component, because by using the VR-Forces graphical user interface (GUI), that gives user a 2D and 3D views of a simulated environment we can observe the interaction between all entities.

3.2.5.1.5 MÄ K Data Logger.

We choose the MÄ K Data Logger for data record and AAR, because the MÄ K Data Logger can provide a way to capture and replay data from the LVC simulations case study, allowing for easy analysis and AAR. Simulation recordings can be zoomed into, edited, and manipulated in a variety of ways.

3.2.5.1.6 MÄ K WebLVC

WebLVC server is an interoperability protocol that enables web-based simulation systems (or federates) to interoperate in M&S simulation environment (or federation). WebLVC client applications using a tablet PC communicates with the rest of the simulation environment (or federation) through an LVC server, which participates in the federation on behalf of one or more clients. The WebLVC protocol defines a standard way of passing simulation data between a web-based client application and an LVC server - independent of the protocol used in the federation. Thus, WebLVC clients can participate in a DIS exercise, an HLA federation, a TENA execution, or other distributed simulation environments (Granowetter, 2013).

3.2.5.2 Step 5-2: Conduct of the Case Study

The case study is an executed, focused experiments of AddSIM, VR-Forces, SIMbox and Data Logger with existing SSAs, (minimal simulation environment (or federation) instantiations) to ensure that the SSAs can be used, gain a better understanding of how each SSA functions, and to assess the relative level of difficulty in instantiating a simulation environment (or federation) using the existing SSAs. Existing gateways or middleware is used to connect the different simulation environments (or federations). Through this execution, a greater appreciation regarding interoperability with the multiple SSAs can be obtained.

3.2.5.3 Step 5-3: Case Study Findings

The LVC simulation case study is analyzed to identify the major problems that exist and to suggest solutions to these problems. In this step, we reported the results of the LVC simulation case study. Then the findings were mapped to requirements for LVC-ITA. Through the mapping between requirements and findings, we identified the problems, and selected main problems that must be resolved for LVC-ITA roadmap.

3.2.5.4 Step 5-4: Case Study Lessons Learned

The identified main problems in Step 5-3 are analyzed and evaluated. We draw lessons learned to solve the problems or limitations from the results of the case study. The lessons learned can help us find the technologies to solve the problems or limitations. Based on the derived lessons learned, possible factors that can improve the LVC simulation environments are explored and utilized for an agile roadmap of the LVC-ITA. M&S communities should keep the lessons learned because lessons learned are key educational components. The lessons learned should help us design an agile roadmap of the LVC-ITA and avoid repeating problems.

3.2.5.5 Step 5-5: Recommended Actions

Recommended actions are to recommend the best solution to be implemented. The lessons learned are intended as recommendations for either improving the current M&S environments or for concepts that should be applied to the agile roadmap for LVC-ITA. The recommended actions to address the needs from the lessons learned were identified by researchers at SIL at UCF. Based on those, we designed the agile roadmap of the LVC-ITA.

3.2.6 Step 6: Agile Roadmap for LVC-ITA

The final road map was developed from discussions with SIL researchers at UCF for an agile roadmap for the LVC-ITA. In this step, we described in detail how these recommended actions should be implemented.

CHAPTER FOUR: CASE STUDY

This chapter describes in detail the case study in six steps. Yin (2014) has defined case study as “an empirical inquiry that investigates a contemporary phenomenon (the ‘case’) in depth and within its real-world context, especially when the boundaries between phenomenon and context may not be clearly evident,” in his book. We considered, conducting case study research would be the preferred method, in this situation when the central research questions are “how” or “why” in Table 6. A single case study can be the basis for significant generalizations of LVC-ITA roadmap.

Ultimately, we want to know how to build the agile roadmap for LVC-ITA. A successful case study analyzes a real-life situation where existing problems need to be solved. Therefore, the objective of the case study is to analyze and evaluate the LVC simulation systems that reflect current M&S technologies. In addition, the case study is to investigate the technologies and methodologies to apply to LVC-ITA from lessons learned. Then, we explain the reason why we choose the technologies among several technologies for LVC-ITA. Publishing our case study and summarizing lessons learned will encourage M&S communities to follow the agile roadmap of LVC-ITA and can help to prevent errors from being repeated.

4.1 Background

The case study was conducted as part of a research project that was realized by the University of Central Florida (UCF) Industrial Engineering & Management Systems (IEMS) Simulation Interoperability Laboratory (SIL). The SIL was responsible for research tasks to develop a sample test bed to demonstrate the interoperable LVC components in a unified simulation environment, and provide technical consulting and technology transfer on ensuring LVC capability in AddSIM.

4.2 Planning a Case Study

This section presents the plan of the case study. The plan of the case study describes the overall process which consists of six phases as shown in Figure 47.

Figure 47: Case Study Process

4.3 Phase 1: Research Questions

The goal of the research is to provide an agile roadmap for the LVC-ITA. In order to achieve this research's goal, the research questions are as summarized in Table 6. These research questions consist of two forms: (a) central question and (b) associated sub-questions.

Table 6: Research Questions

Area	Questions
Central Questions	• How can we develop a seamless LVC simulation environment? • What technologies are needed to execute a successful LVC simulation?
Associated Sub-questions	• What are the problems with the current LVC simulation? • How to find the problems? • How to solve the identified problems? • What is the latest Modeling and Simulation (M&S) technology?

4.4 Phase 2: Designing a Case Study

This section describes the components and incremental steps for the LVC simulation case study. We developed a case study design in stages. If the previous step succeeds, it may proceed to a more advanced design stage. In the case study, a LVC simulation configuration was defined to create *Air Defense Engagement* scenarios.

In the first step, we built a federation using only HLA as shown in Figure 48. Figure 48 depicts the design of the Air Defense Engagement simulation environment (or federation). The HLA target federation consists of five simulation systems, including two Virtual simulators, Constructive simulation, a component based simulation environment (AddSIM), and Data Logger for After Action Review (AAR). The following subsections describe each component simulation system in detail.

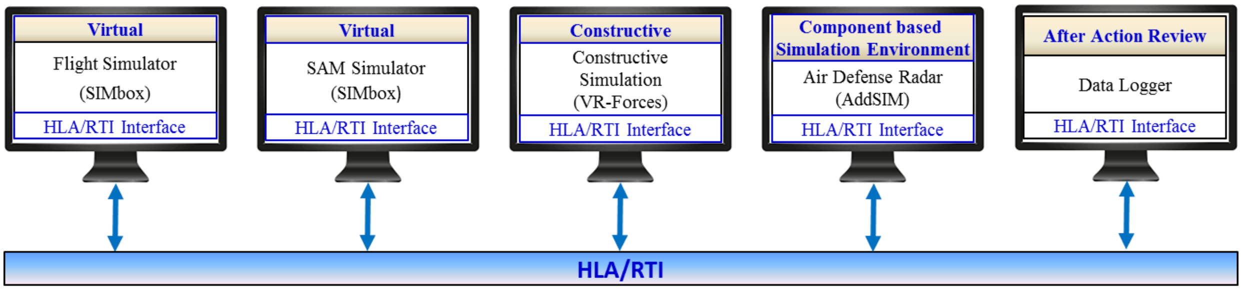


Figure 48: HLA Federation for Air Defense Engagement

In the second step, we connected the HLA based target federation via WebLVC server to a tablet PC as shown in Figure 49. A tablet PC as Live component was used in order to interact, through a WebLVC server with the Constructive and Virtual components in the below the framework. Target federation can be shown and operated in the tablet PC.

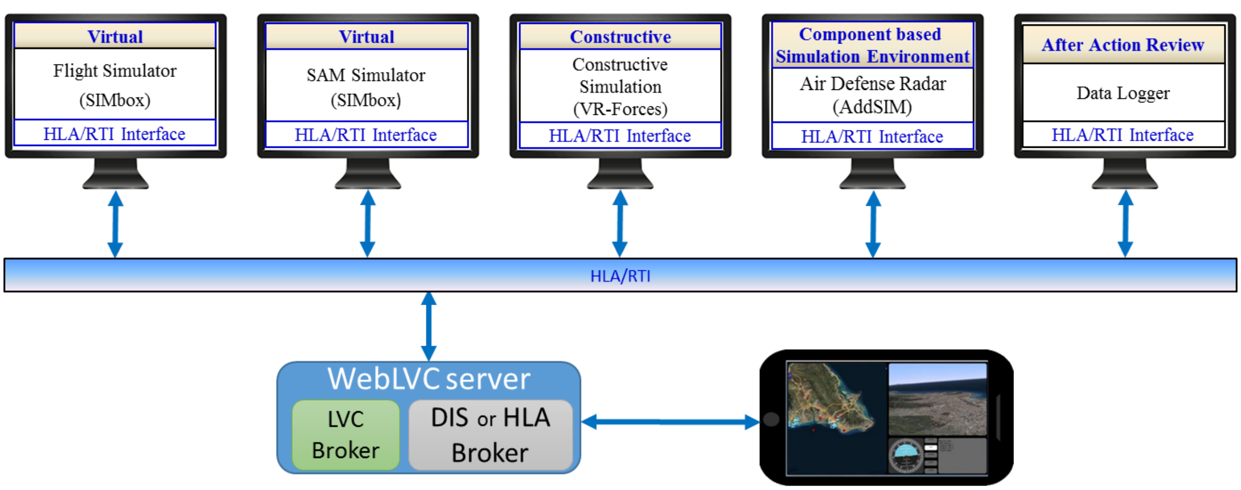


Figure 49: HLA Federation with WebLVC for Air Defense Engagement

In the final step, the LVC distributed simulation configuration was based on the DIS and HLA with a target simulation environment (or federation). The Air-Defense Engagement federation is consist of two federations. The one federation is the DIS based federation for Flight Simulator and SAM Simulator. The other federation is the HLA based federation for Constructive Simulation, Air Defense Radar of AddSIM and Data Logger. We connected DIS based federation and HLA based federation with WebLVC server as shown in Figure 50.

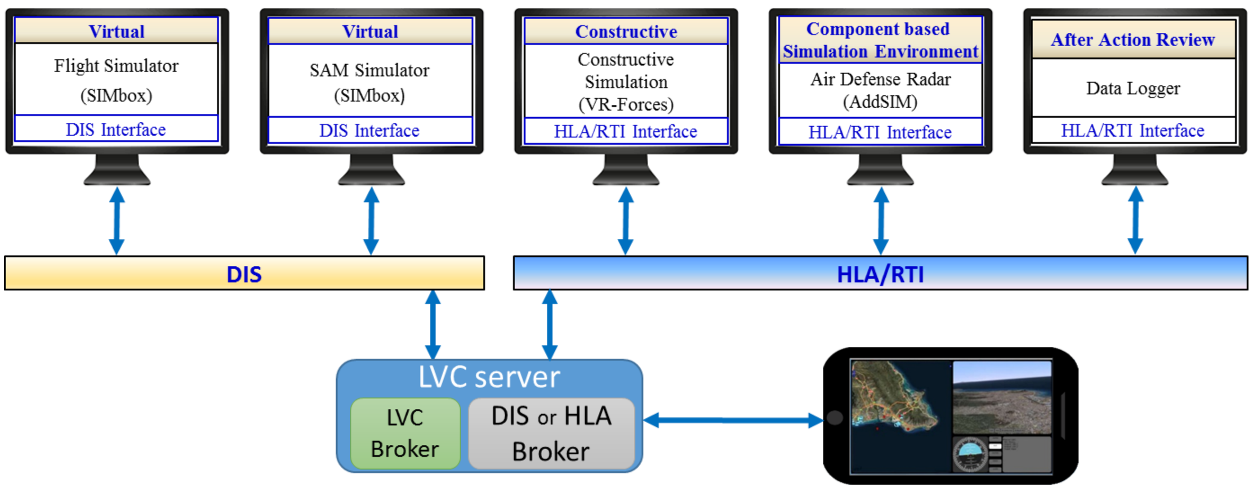


Figure 50: Final Design for the LVC simulation case study

4.4.1 Component-based simulation environment: Adaptive distributed parallel simulation environment based on interoperable and reusable models (AddSIM)

This section describes the architecture and operation concept of the Adaptive distributed parallel simulation environment based on interoperable and reusable model (AddSIM) which is a component-based simulation environment for integrated M&S systems. This simulation environment makes it possible to search and use component-type models stored in local or remote resource repositories, which enables users to assemble or reconfigure models depending on the user's purpose by plug-in and easy play style.

4.4.1.1 Overview

AddSIM that has been developed by Agency for Defense Development (ADD) in South Korea is a component-based simulation environment. The first version of AddSIM was developed through a core technology R&D project of ADD from 2009 to 2011. The main goal of AddSIM is to enhance interoperability, reusability, and composability of weapon simulation models. In order to improve the reusability, interoperability, and composability of simulation systems, the concept to separate a model from a simulation engine was applied to AddSIM (Kim et al., 2013).

4.4.1.2 Architecture of AddSIM

AddSIM was designed in the layered architecture for prevention against duplication of functions at each layer, ease of maintenance and convenience in developing models as shown in Figure 51. Furthermore, it was designed in the form of simulation architecture using shared memory based on middleware to increase the real-time processing capability of the simulation. In order to do this, the Tao- Common Object Request Broker Architecture (Tao-CORBA) is used as a middleware and multi passing interface (MPI) concept for parallel distributed processing of the simulation is applied (Lee et al., 2012).

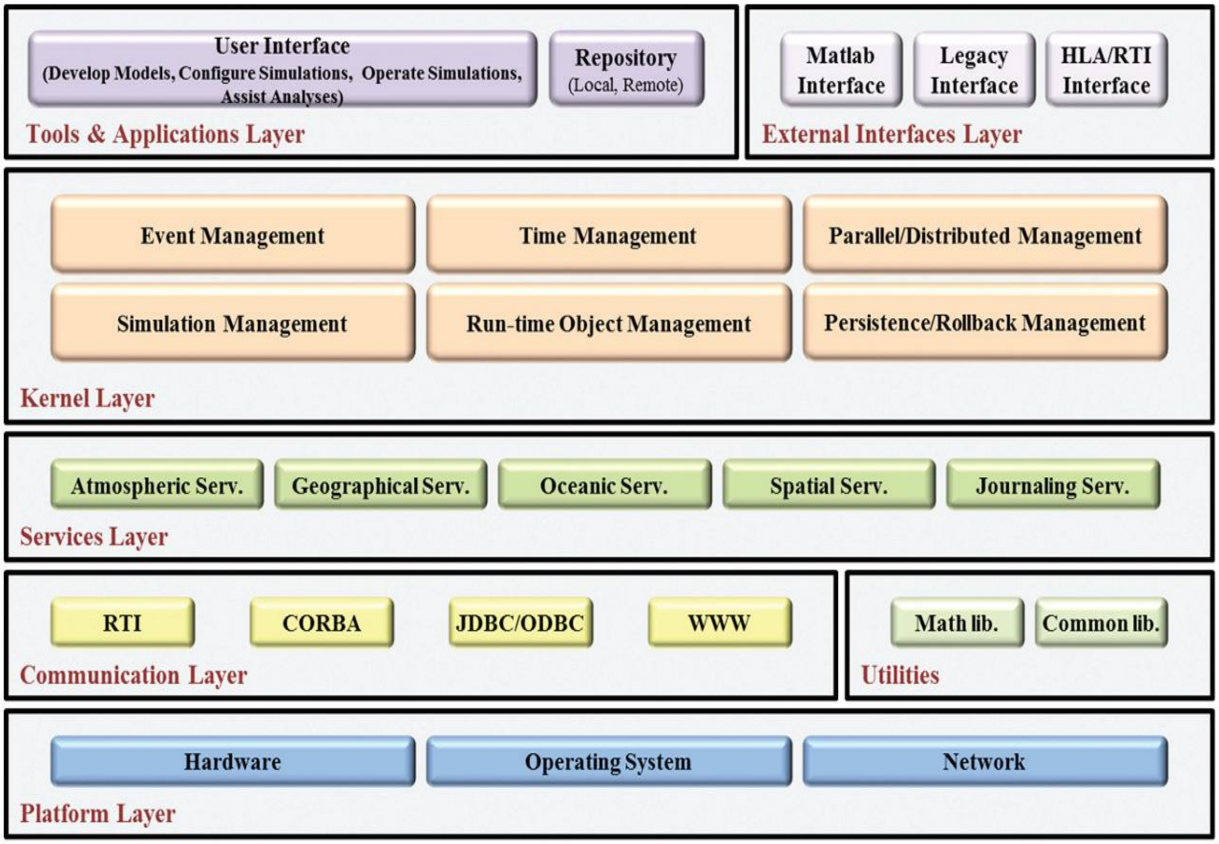


Figure 51: Layered Architecture of AddSIM

Source: Lee et al. (2012)

The architecture consists of a tool & application layer, external interfaces layer, kernel layer, service layer, communications layer, and platform layer.

<u>4.4.1.2.1 Tool and Application Layer</u>

In a tool and application layer, component & player development, build & execution, and analysis of simulation, search and use of componentized models in distributed repositories are performed. The graphical editing framework (GEF) based on Eclipse is used as a development tool to increase the user convenience and efficiency of the components and player development. To support the reuse of components, an editing tool provides properties of components in EXtensible Markup Language (XML) format. The standard structure of component is referred to as Base Object Model (BOM) of SISO.

The web server for component model is linked with the xml file automatically when the component is shared. During the time the component is developed, the xml file that is used in the simulation configuration and operation for the model is made. AddSIM also provides the post-analysis module to analyze the simulation result and visualization module using SIMDIS 3-D Analysis and Display Toolset to play back the entire simulation execution (Lee et al., 2012).

4.4.1.2.2 Kernel Layer

Kernel layer that is a core layer of AddSIM consists of six functions, including parallel and distributed management for parallel processing in distributed environment as well as the five basic functions of event management; time management and simulation management, run-time object management and persistence & rollback management. The Procedure for executing the simulation in kernel layer is as follows. After loading componentized models stored in a local and remote repository based on created simulation file in tool & application layer, simulation object is created. Then, run-time objects of simulation are executed. After that, the kernel processes simulation events, which is communication with other runtime simulation objects through messages, stores properties of simulation objects and conducts relay of service for a service layer (Lee et al., 2012).

4.4.1.2.3 Service Layer

Service layer supports APIs for the high-fidelity models. Users can easily describe the weapon system by using environmental APIs of atmosphere, ocean, and geography.
The atmospheric and oceanic APIs is designed to treat the meteorological data format such as, GRIdded Binary (GRIB), Synthetic Environment Data Representation and Interchange Specification (SEDRIS) transmittal format (STF) and Network Common Data File (NetCDF) through transforming data into ASCII files. The geographical API is designed to handle the flat and ellipsoidal earth model as well as to manage the Digital Terrain Elevation Data (DTED) and Feature Database (FDB) format to extract the geographical feature. User can handle the simulation object's spatial information such as position, speed, and user defined data. Journaling API saves and extracts log data generated during the simulation execution and user defined variables (Lee et al., 2012).

4.4.1.2.4 External Interface Layer

In terms of the external interface layer, there are many simulation resources developed with C and C++ or Matlab in military simulation. Also, many simulation resources are federated through HLA/RTI. HLA is a de-facto SSA for now, and HLA compliancy is a necessary condition to meet current simulation environment requirements. Therefore, simulation environment has to support the interoperability with these legacy simulation resources to enhance the reuse of simulation. For these reasons, AddSIM provides three external interfaces such as C, C++, Matlab, DIS and HLA/RTI interface (Lee et al., 2012).

4.4.1.3 Features of AddSIM

AddSIM has several distinguishing features compared to existing conventional simulation environments.

<u>4.4.1.3.1 Separation between a Simulation Engine and Models</u>

The first of the distinguishing features is the separation between a simulation engine and models. Modeling framework in AddSIM has been developed upon *Open Simulation Architecture for Modeling and Simulation* (OSAMS) that is being studied as an open modelling framework in Parallel and Distributed Modeling & Simulation Standing Study Group (PDMS-SSG) of Simulation Interoperability Standards Organization (SISO) and Base Object Model (BOM), SISO standard for simulation object model (J. Steinman & Parks, 2007).

<u>4.4.1.3.2 Standardization of a Modeling Framework</u>

The second feature is the standardization of a modeling framework. A simulation model is designed to have a hierarchical structure as shown in Figure 52. The top level is the simulation model that includes some players. Each player consists of some components. Furthermore, each component can include sub-components recursively.

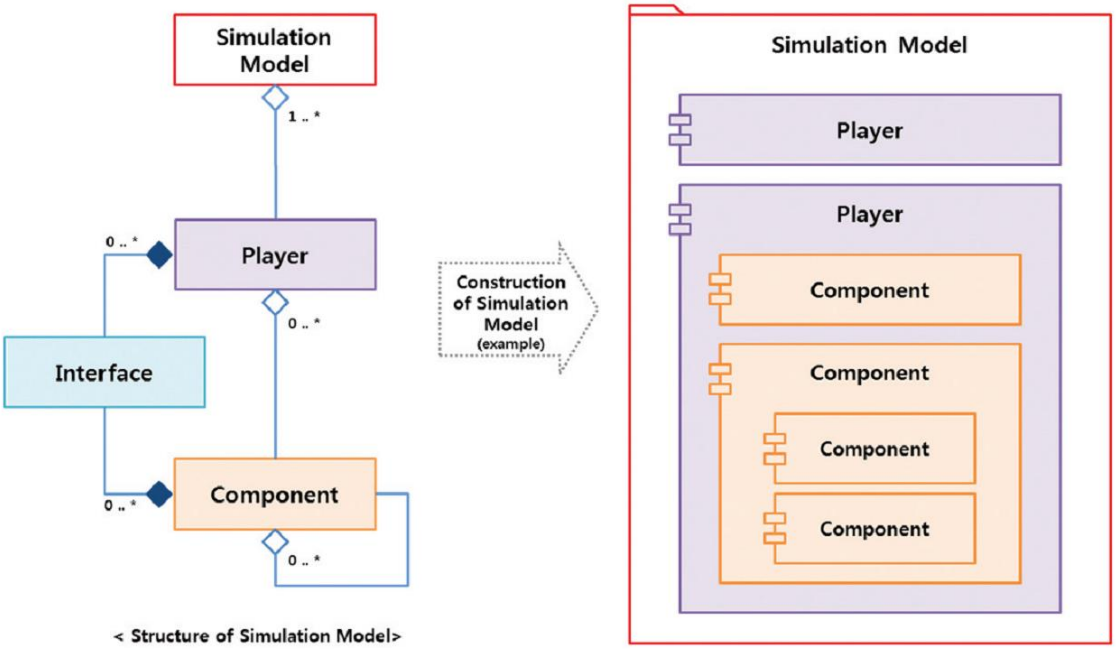


Figure 52: A hierarchical modeling structure of AddSIM

Source: Lee et al. (2012)

The definition of the, player, component and interface is as follows (Lee et al., 2012).

● Player: It is the top level component model configuring the simulation model. Usually, it represents a weapon system such as flight, tank or missile. The behavior of a player is modeled with a user defined code (UDC).

● Component: It is a building block (an element of a player or upper component) that executes a specific function independently. The behavior of an element is also modeled with a UDC. A component is compiled into a dynamic link library (DLL) and linked with AddSIM.

● Interface: It is a passage to process events of kernel, components and players. Components and players via the interface can communicate each other.

In the modeling procedure, common meta model is used to improve interoperability and reuse of the model. AddSIM also uses meta model for component and player modeling. In the AddSIM, meta-model defines the relationship between component, player, interface, member function, variable, and data type. Using the hierarchical structure and common meta model for component and player, AddSIM can enhance interoperability and reuse of components and players. Components and players are compiled by way of componentizing to configure the dynamic loading for simulation. Meta-information for a component such as configuration information, communication information, and control information is stored and controlled in XML style. While a simulation is executed, a kernel interprets that file for configuring simulation objects. As AddSIM provides dynamical loading of simulation objects, components stored in remote repositories are retrieved or used without any modification of components by downloading.

4.4.1.3.3 Web Service based on SOA Concept

The third characteristic is web service based on SOA concept. To support distributed simulation smoothly, the distributed resource repository based on web is provided. Through the web service, users can retrieve and reuse components stored in a remote repository. Figure 53 shows the operational concept of distributed repository.

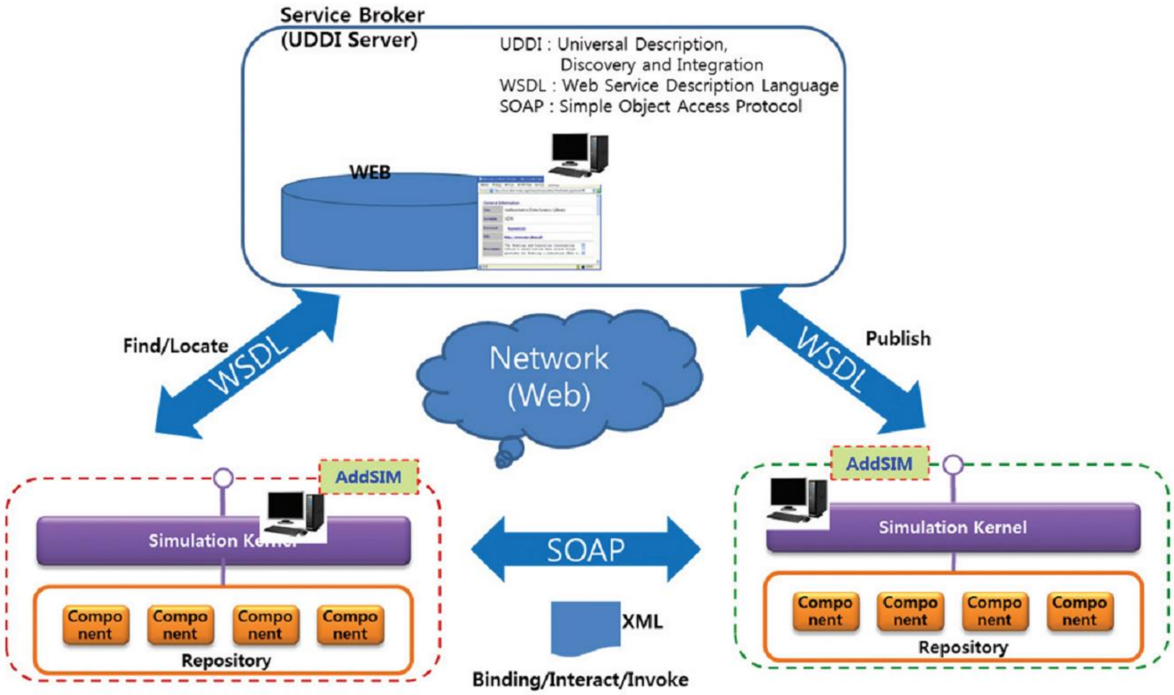


Figure 53: Operational concept of distributed repository.

Source: Lee et al. (2012)

4.4.1.3.4 Time Synchronization Algorithm

Finally, AddSIM engine provides the infrastructure and related functions capable of working number of event processes and synchronizing time between event processes in order to do parallel processing at the same time. Time synchronization algorithm for parallel processing can be divided into a conservative and optimistic way. In the optimistic way, there are time warps, breathing time bucket (BTB), breathing time warp (BTW), etc. Among the optimistic way, AddSIM engine is designed to utilize BTB algorithm and rollback handling for time synchronization between event processes when proceeding parallel processing. In BTB algorithm, each process broadcasts the oldest local, even among those it will execute. This is called a local event horizon (LEH). A process must suspend its even processing if it has received an older LEH than the one it is currently processing. The oldest LEH among all processes becomes the next global event horizon (GEH). Each process may send out all messages and processes all events before this new GEH. Processes which have already processed beyond GEH must roll back their computation to GEH. No anti-messages are sent out (Lee et al., 2012).

AddSIM engine offers the infrastructure and related functions capable of generating runtime objects located in a remote place and passing the interaction messages between runtime objects. All constituents of the kernel are operated based on CORBA. Management of runtime object located in remote place is performed by remote kernel, but event management is performed by master kernel through the configuration of the constituent information when kernels are connected.

4.4.2 Virtual Simulator: SIMbox

This section presents the SIMbox Virtual simulator. We developed a simulation system of Flight and Surface-to-Air-Missile (SAM) in SIMbox simulation platform that is a Commercial off the Shelf (COTS) simulation system. SIMbox is a software platform and a distributed simulation solution for defense and civilian applications. SIMbox concept is a set of development tools for components based design and creation. SIMbox uses solution software for content creation, simulation, visualization, human-machine interface and graphics modeling tools. SIMbox contains several software modules empowering users or developer in creating new contents and environments. Figure 54 shows the detailed interior, exterior and weapons of the F-16 flight model.

Cockpit	Exterior
	
AIM-9 Air to Air Missile	MK-84 Air to Ground Bomb
	

Figure 54: F-16 Flight Simulator

The SA-8 SAM entity was implemented using SIMbox Toolkit. We developed or modified the SA-8 SAM model and the cockpit of SA-8. There are five main functional features we developed for SA-8 SAM entity:

- Switches, Buttons and Knobs

● Electrical System

● Weapon Control and Display

● Search and Track Radar

● Warning Sounds

We also developed SAM RADAR screen using Console Editor. The radar screen demonstrates the ID, Target altitude label (ALT), Air Speed label (SPD), heading label (HDG), Distance label (DIST) and Aspect ratio label (ASP) of the primary target. Therefore, SAM radar has all the labels for the primary target data. Figure 55 shows the interior, exterior and radar of the SA-8 SAM simulator.

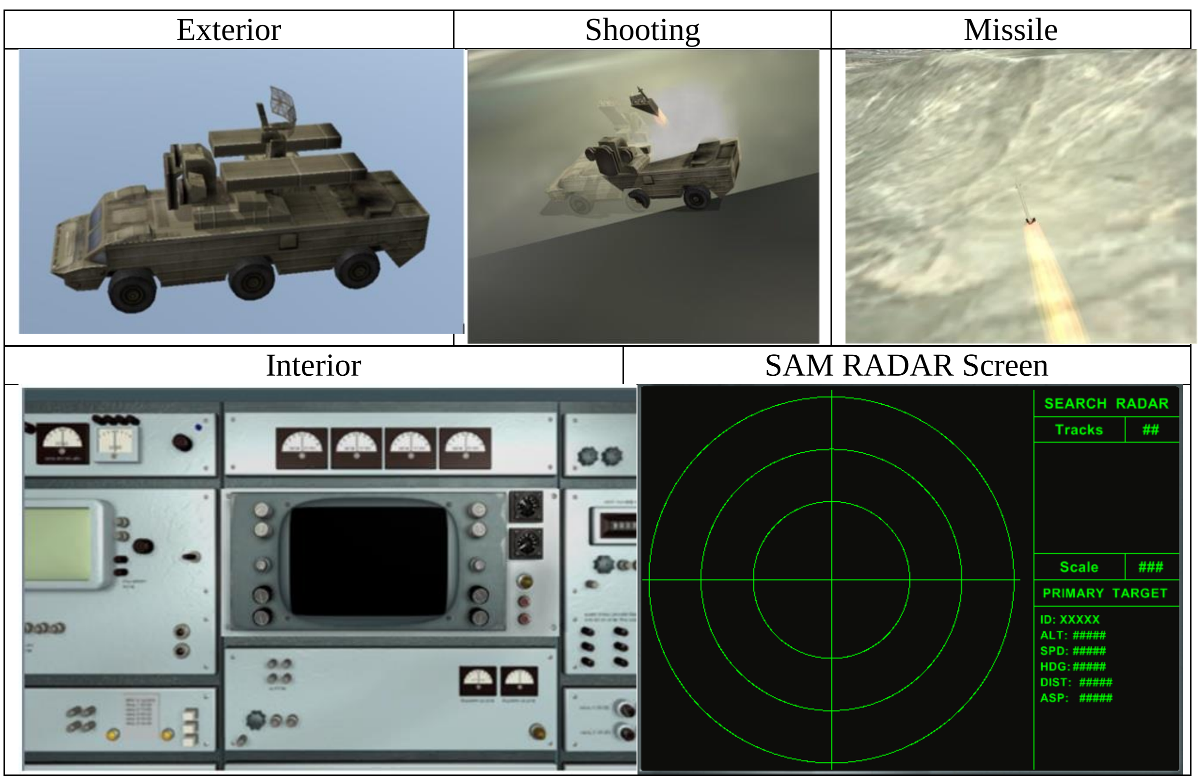


Figure 55: SA-8 SAM Simulator

4.4.3 Constrictive Simulation System: VR-Forces

The main COTS tool used is MÄ K VR-Forces, a powerful and flexible simulation environment for scenario generation. It has all the necessary features for developing Computer Generated Forces (CGF) for simulating a complex operational environment.

VR-Forces Computer Generated Forces provides 3D and 2D views of your simulated world, integrated into one graphical user interface (GUI) that allows non-programmers to build scenarios by positioning forces, creating routes and waypoints, and assigning tasks or plans with a simple point and click. User can place icons on a 2D tactical map for large scale scenario development, or drag and drop human entity models directly into a 3D scene to accurately position them inside of buildings or behind trees. During scenario execution, VR-Forces vehicles and human entities interact with the terrain, follow roads, avoid obstacles, communicate over simulated radios, detect and engage enemy forces, and calculate damage VR-Forces comes with simulation models for a wide variety of battlefield entities and weapon systems (MÄ K).

Some useful features of VR-Forces are:

- includes a C++ toolkit to extend or embed VR-Forces in another computer application
- can be used as distributed simulation engine with remote GUI control
- can aggregate unit and entity modeling
- supports standard simulation protocols such as HLA and DIS
- supports various kinds of terrain, including streaming terrain
- supports GUI-based entity and parameter editing

4.4.4 WebLVC Server

WebLVC server is an interoperability protocol that enables web-based application to interoperate in M&S federations. WebLVC client applications using a smartphone or tablet PC communicate with the rest of the federation through a WebLVC server, which participates in the federation on behalf of one or more clients. The WebLVC protocol defines a standard way of passing simulation data between a web-based client application and a WebLVC server - independent of the protocol used in the simulation environment (or federation). Thus, a WebLVC client can participate in a DIS exercise, an HLA federation, a TENA execution, or other distributed simulation environments (Granowetter, 2013).

The WebLVC protocol specifies a standard way of encoding object update messages, interaction messages, and administrative messages as JavaScript Object Notation (JSON) objects, which are passed between client and server using WebSockets. LVC server is flexible enough to support representation of arbitrary types of objects and interactions (i.e. arbitrary Object Models). However, WebLVC server does include a Standard Object Model definition based on the semantics of the DIS and HLA's RPR FOM (Granowetter, 2013). Users can extend the Standard Object Model by adding new types of objects, attributes, interactions, and parameters; or can choose to represent the semantics of entirely different Object Models (e.g. other HLA FOMs, Architecture Neutral Data Exchange Mode (ANDEM) models, etc.) Live component can describe a commander and instructor. Commander can command the entities from a tactical map interface.

4.4.5 Data Logger

The MÄ K's Data Logger is a system for capturing and replaying simulation data. The MÄ K's Data Logger can record HLA and DIS messages and replay them back for After-Action Review (AAR) and analysis. A recorded file can be fast forwarded or played in slow motion, and areas of interest located quickly. The MÄ K's Data Logger provides the Graphic User Interface (GUI) that allows user to visually edit the simulation recording (MÄ K).

4.5 Phase 3: Conducting a Case Study

This section describes the conduct of the case study in detail.

4.5.1 Objective

The objective of the LVC simulation case study is to verify the LVC simulation interoperability by demonstrating Air-Defense Engagement between Virtual Flight simulator and Virtual SAM simulator in SIMbox, Constructive VR-Forces simulation system, Constructive engineering level Air Defense Radar player in AddSIM and Data Logger using HLA/RTI and DIS external interface. In addition, the case study is to verify the interaction of Live component through WebLVC server.

4.5.2 Member Applications

The HLA/RTI and DIS simulation environments consist of five simulation systems: F-16 flight simulator (SIMbox), SA-8 SAM (SIMbox), Constructive Simulation (VR-Forces), Air

Defense Radar (AddSIM) and AAR (Data Logger). Figure 56 shows the players (or entities) in the case study.

Figure 56: Plyers (or Entities) in the case study

Figure 57 describes overview of the hardware and software specification in the case study environment.

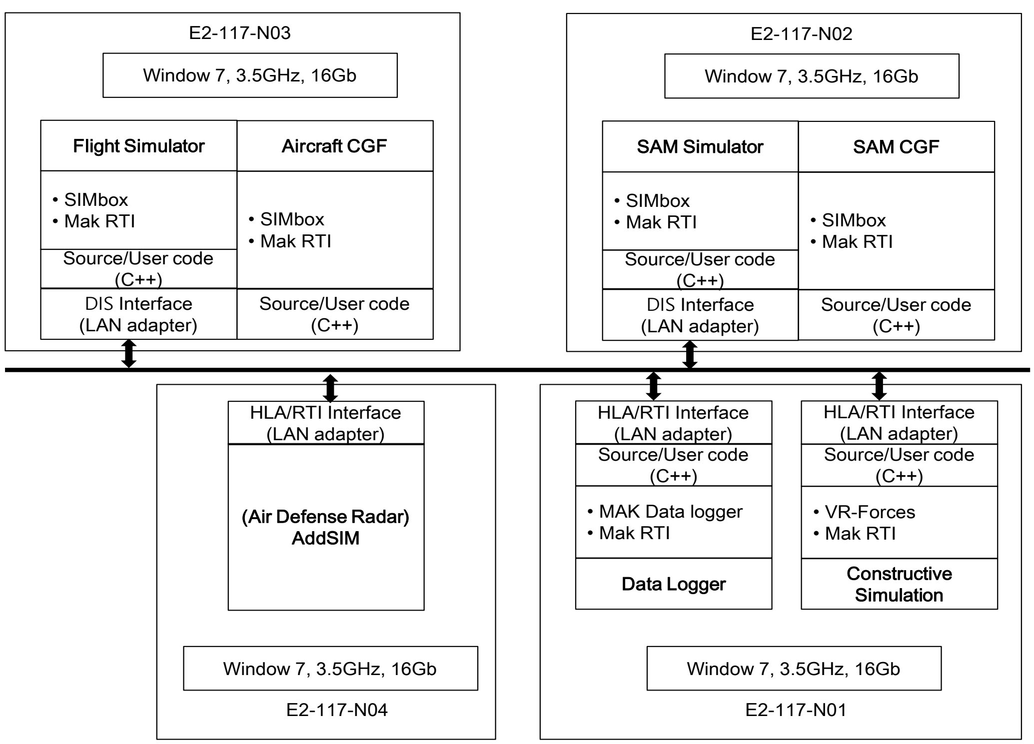


Figure 57: Hardware and Software Specification of the Case Study Environment

Table 7, 8, 9 and 10 describes the operation environment of each simulation system (or federate) in the case study.

Table 7: Virtual Flight Simulator

Content	Equipment	Description
Operation Environment	Desktop Computer	• CPU: Intel(R) Core(TM) i7-4770K Processor 3.50GHz • Memory : 16GB • HDD : 1TB • ODD : DVD-Multi • VGA : NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD (1920x1080)
	O/S	• Window 7
	Operation	• SIMbox Development Toolkit • MÄK RTI
	Complier	• Microsoft Visual Studio 2010

Note. CPU= Central Processing Unit, HDD = Hard Disk Drive, ODD = Optical Disc Drive, VGA= Video Graphics Array, DVD = Digital Video Disc, LCD = Liquid Crystal Display

Table 8: Virtual SAM Simulator

Content	Equipment	Description
Operation Environment	Desktop Computer	• CPU: Intel(R) Core(TM) i7-4770K Processor 3.50GHz • Memory : 16GB • HDD : 1TB • ODD : DVD-Multi • VGA : NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD (1920x1080)
	O/S	• Window 7
	Operation	• SIMbox Development Toolkit • MÄK RTI
	Complier	• Microsoft Visual Studio 2010

Table 9: Constructive Simulation and Data Logger

Content	Equipment	Description
Operation Environment	Desktop Computer	• CPU: Intel(R) Core(TM) i7-4770K Processor 3.50GHz • Memory : 16GB • HDD : 1TB • ODD : DVD-Multi • VGA : NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD (1920x1080)
	O/S	• Window 7
	Operation	• MÄ K VR-Forces • MÄK RTI • MÄK Data Logger
	Complier	• Microsoft Visual Studio 2010

Table 10: AddSIM

Content	Equipment	Description
Operation Environment	Desktop Computer	• CPU: Intel(R) Core(TM) i7-4770K Processor 3.50GHz • Memory : 16GB • HDD : 1TB • ODD : DVD-Multi • VGA : NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD (1920x1080)
	O/S	• Window 7
	Operation	• AddSIM • MÄK RTI
	Complier	• Microsoft Visual Studio 2010

4.5.3 Prerequisite Condition

First, HLA/RTI and DIS should be set up in the network. Second, F-16 flight simulator and SA-8 SAM simulator developed by UCF are linked via DIS and operated. Third, Air Defense Radar player in AddSIM, VR-Forces simulation and Data Logger are linked via HLA/RTI and worked. Finally, WebLVC server should be set up to connect the target Air Defense Engagement simulation environment.

4.5.4 Designing Air Defense Engagement Scenario

4.5.4.1 Air Defense Engagement Scenario

The Air Defense Engagement scenario is as shown in Figure 58. The scenario is as follows. First, High-Altitude Air Defense Radar of AddSIM detects the approaching F-16 flight's location. Second, as soon as High-Altitude Air Defense Radar detects the F-16 flight, the Air Defense Radar sends detection information to SA-8 SAM of SIMbox. Third, SA-8 SAM also detects the latest F-16 flight's location and calculates the estimated F-16 flight's position with detecting information and homing guide point for a missile, then fires anti-air missiles. The missile gets the homing guide point and launching signal from SA-8 SAM. The missile flies to the homing guide point with the inertial guide algorithm. After it reaches there, it uses seeker to search and track the F-16 flight. The ending condition is that the distance between the missile and the F-16 flight is within a specified threshold range. And lastly, F-16 flight is destroyed by anti-air-missile.

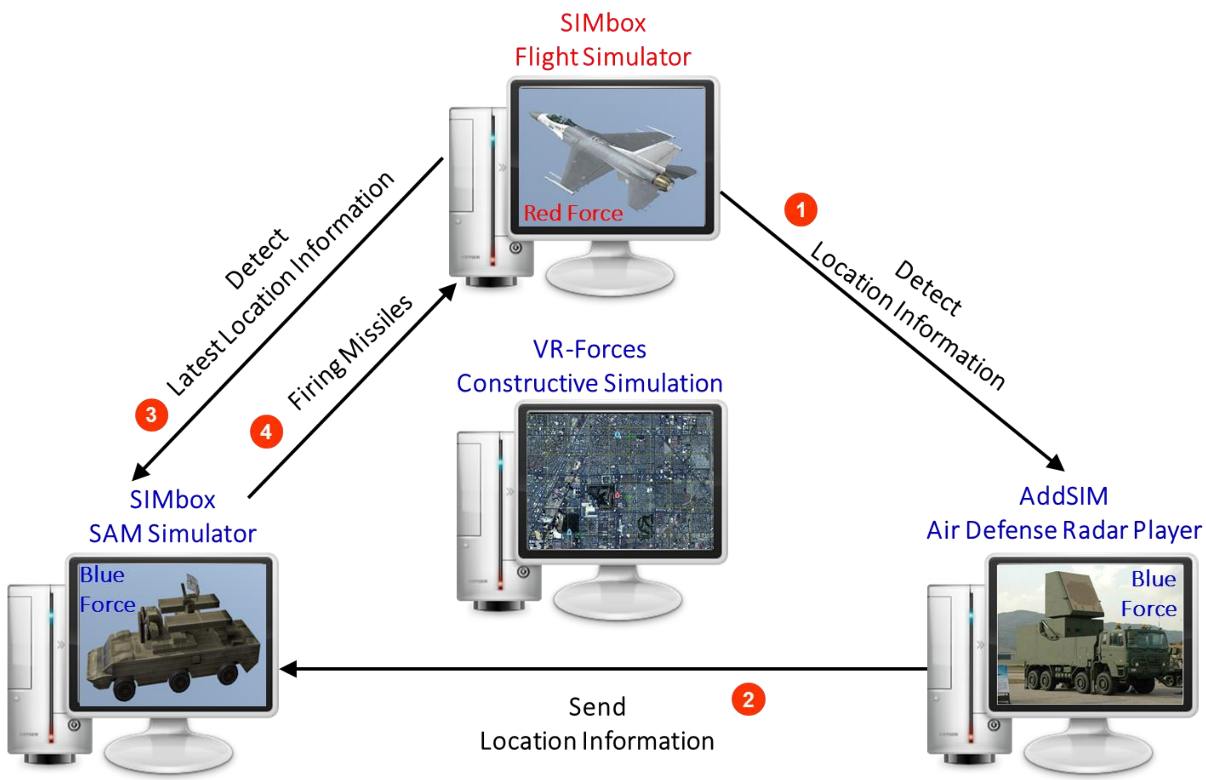


Figure 58: Air Defense Engagement Scenario

4.5.4.2 Natural Environment Condition

- Geographic Area: Las Vegas, Nevada
- Climate: normal daytime
- Simulation Time: March 00, 2015 from 14:00 E.T. until simulation ends.
- Simulation End Condition: F-16 flight is destroyed

Figure 59 shows the geographical condition and initial scenario setting on VR-Forces GUI.

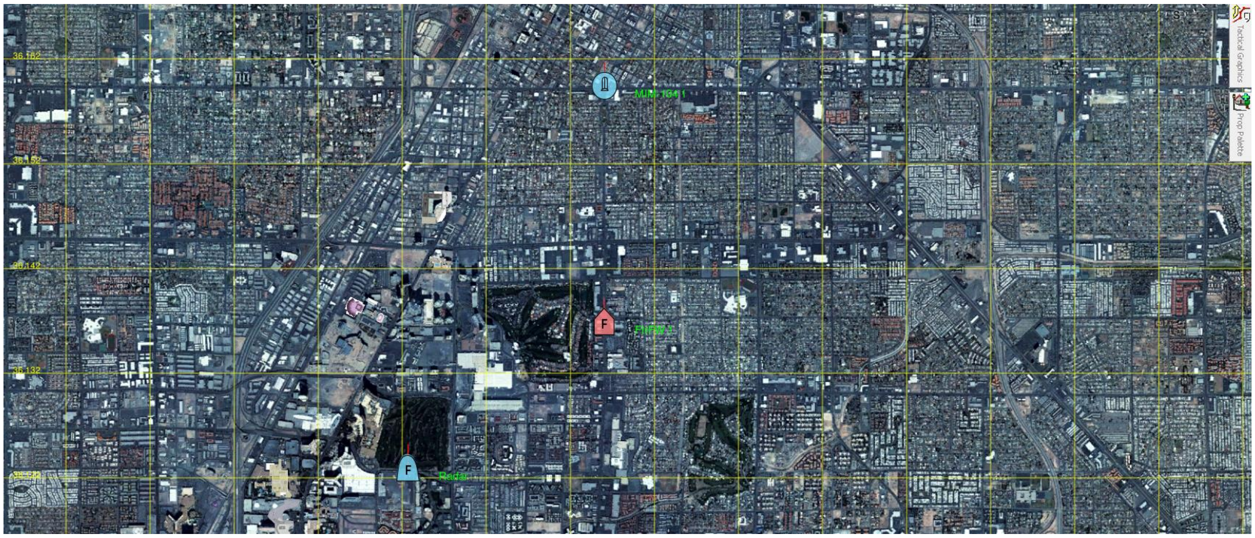


Figure 59: Geographical Condition and Initial Scenario Setting on VR-Forces GUI

4.5.4.3 Expected Test Result

Behavior of Air Defense Radar in AddSIM, F-16 flight and SA-8 SAM in SIMbox are verified on MÄ K RTI. Air Defense Radar in AddSIM sends the F-16 flight detection information to the SA-8 SAM simulator. Then, result of F-16 flight’s evasion or hit from SA-8 SAM attack is provided in SIMbox and VR-Forces. Representation of the engagement result on the VR-Forces, SIMbox and Tablet PC is provided.

4.5.5 Procedure of Air Defense Engagement Simulation

This section describes the overall procedure of Air Defense Engagement Simulation.

First, open the Virtual-Virtual (VV) simulation environment (or federation) of F-16 flight simulator and SA-8 SAM simulator in DIS and Constructive-Constructive (CC) simulation environment (or federation) of VR-Forces, Air Defense Radar and Data Logger in HLA for Air Defense Engagement Simulation.

Second, check the all the simulation systems (or federates) on MÄ K RTI and, set up the initial position of F-16 flight entity and SA-8 SAM entity in SIMbox and Air Defense Radar player entity in AddSIM for detecting the F-16 flight as shown in Figure 60 and 61.

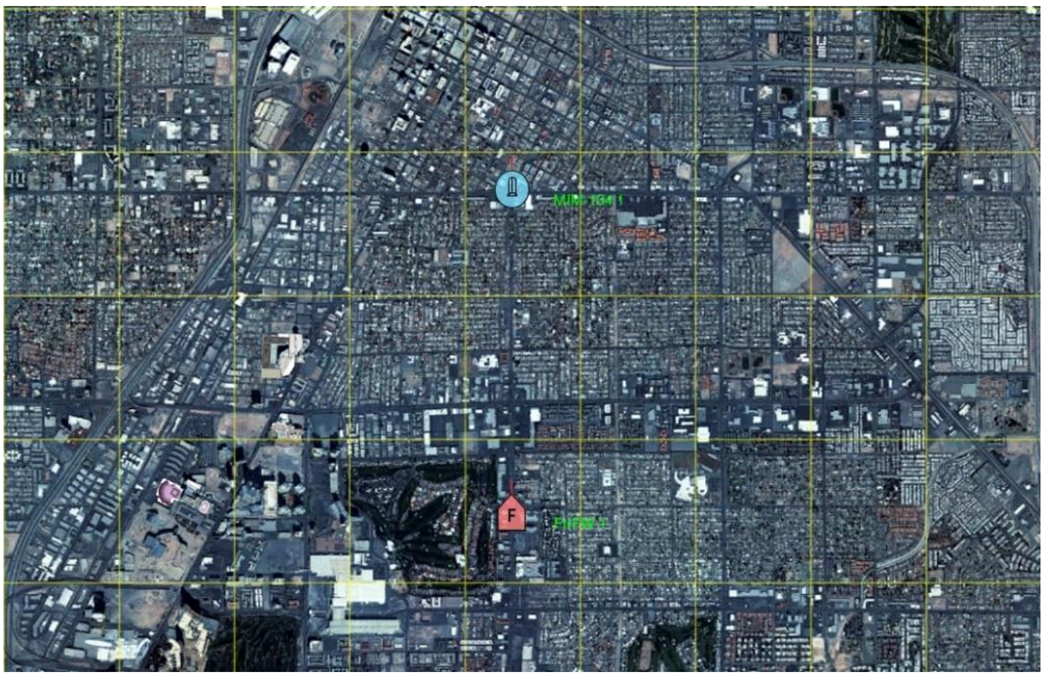


Figure 60: Initial Situation of SA-8 SAM (Blue force) and F-16 Flight (Red force)

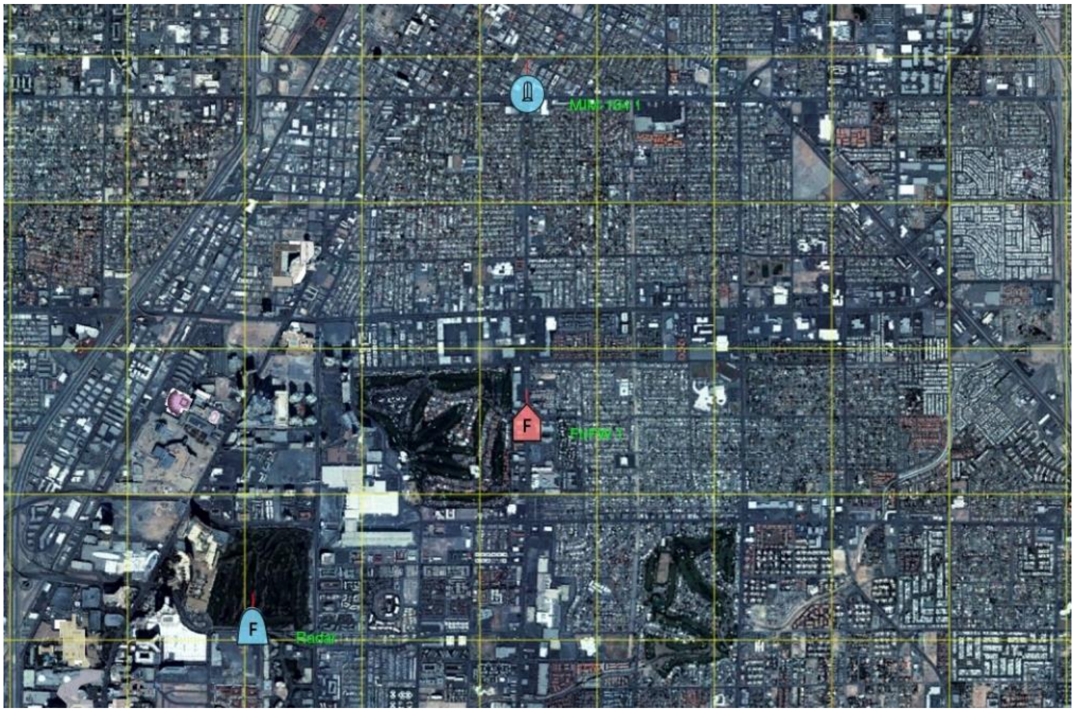


Figure 61: Join of AddSIM's Air Defense Radar Player from Initial Situation

Third, check whether the coordinates are consistent for all entities as shown in Figure 62 and 63. Comparing the two pictures, the coordinates can be seen that a slight discrepancy. This issue is discussed in Section 4.6.7.

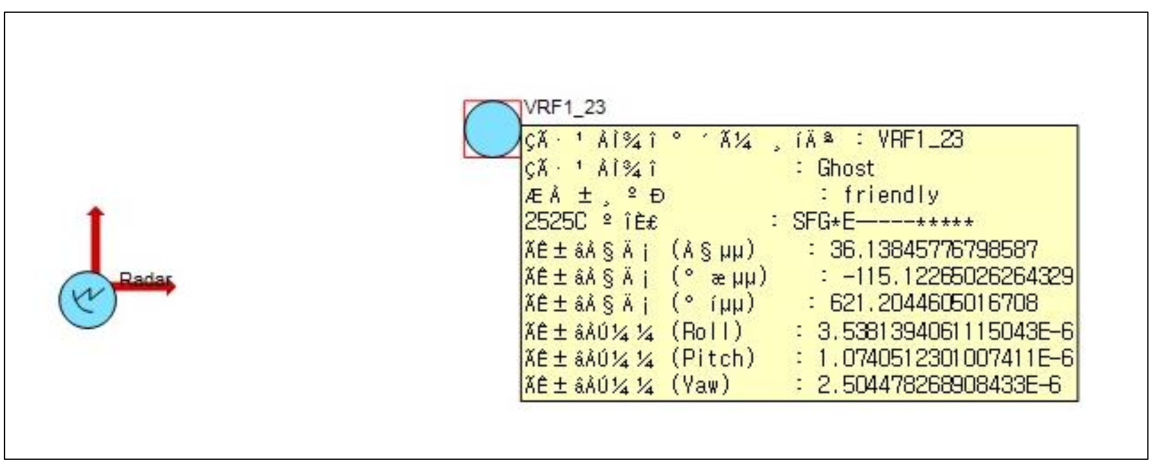


Figure 62: F-16 Flight Information in AddSIM

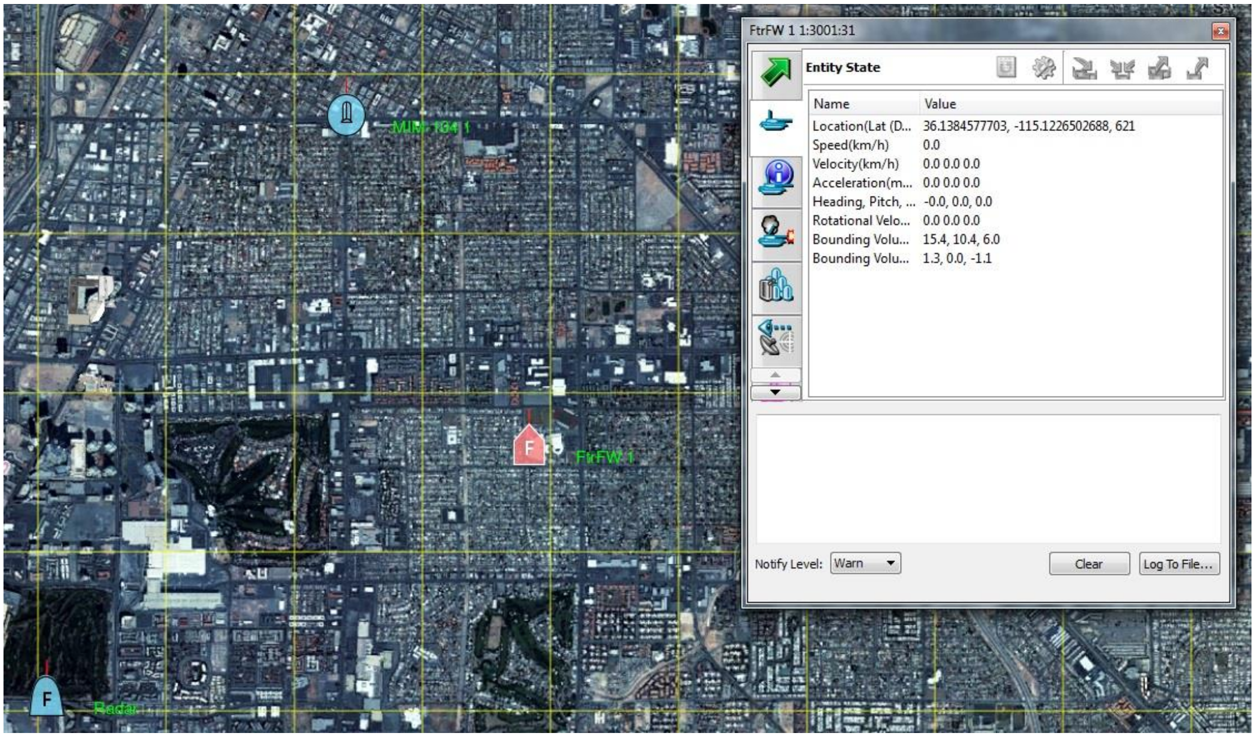


Figure 63: F-16 Flight Information in VR-Forces

Fourth, in turn, execute each federates. In order words, execute Air Defense Radar in AddSIM, F-16 flight simulator and SA-8 SAM simulator in SIMbox, VR-Forces simulation to observe all the entities, Data Logger to record the Air-Defense Engagement simulation, and WebLVC server to display on the Web browser.

Fifth, after the execution of each federates, the Air-Defense Engagement simulation is automatically progressed with time. F-16 flight moves within the area that SA-8 SAM and Air Defense Radar located. We check if Air Defense Radar player of AddSIM detects the F-16 flight, and then it sends detection information to SA-8 SAM simulator. Then check if the F-16 flight can be displayed on the screen of SAM simulator. Next, check if the SA-8 SAM simulator attacks F-16 flight.

Lastly, each simulation system calculates Battle Damage Assessment (BDA) as soon as the F-16 flight is hit. I check to see if the F-16 flight that was hit is displayed on VR-Forces GUI as shown in Figure 64.

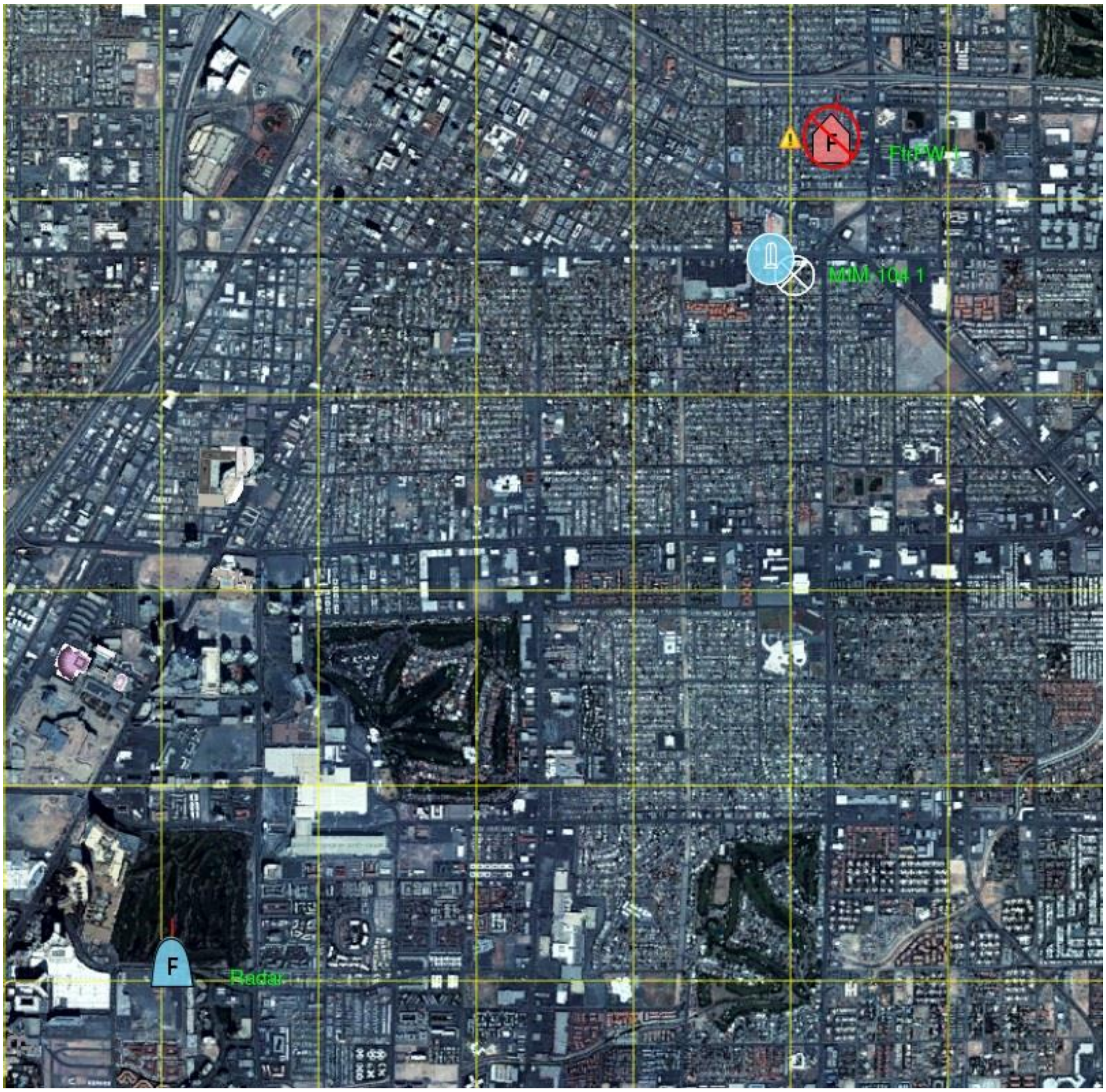


Figure 64: SA-8 SAM's Attack to F-16 flight

4.5.6 Simulation Result Analysis

4.5.6.1 Data Analysis

We checked the Data Logger file for After Action Review (AAR) as shown in Figure 65. We checked that Air Defense Radar player successfully sent the detection information to SA-8 SAM Simulator.

```
*************************************************************************************
**** Packet #11576  Size=256    time=0:04:04.8603, 18:26:09.8203 Thu Mar 26, 2015
******** Type=Interaction #3          *******************************************
Interaction Class Name(Handle): HLAinteractionRoot.WeaponFire(77)
Parameter Count: 13
Parameter #1   Size:    9, Name(Handle): EventIdentifier(0)
Value:
00 2 56 52 46 31 3a 32
00
Parameter #2   Size:    4, Name(Handle): FireControlSolutionRange(1)
Value:
00 0 0 0
Parameter #3   Size:    4, Name(Handle): FireMissionIndex(2)
Value:
00 0 0 0
Parameter #4   Size:   24, Name(Handle): FiringLocation(3)
Value:
00 0 0 0 0 0 0 0
00 0 0 0 0 0 0 0
00 0 0 0 0 0 0 0
Parameter #5   Size:    7, Name(Handle): FiringObjectIdentifier(4)
Value:
56 52 46 31 3a 32 0
Parameter #6   Size:    2, Name(Handle): FuseType(5)
Value:
00 0
Parameter #7   Size:   12, Name(Handle): InitialVelocityVector(6)
Value:
00 0 0 0 0 0 0 0
00 0 0 0
Parameter #8   Size:    1, Name(Handle): MunitionObjectIdentifier(7)
Value:
00
Parameter #9   Size:    8, Name(Handle): MunitionType(8)
Value:
02 1 0 e1 1 10 2 0
Parameter #10  Size:    2, Name(Handle): QuantityFired(9)
Value:
00 1
Parameter #11  Size:    2, Name(Handle): RateOfFire(10)
Value:
00 0
Parameter #12  Size:    7, Name(Handle): TargetObjectIdentifier(11)
Value:
56 52 46 31 3a 31 0
Parameter #13  Size:    2, Name(Handle): WarheadType(12)
Value:
00 0
*************************************************************************************
```

Figure 65: Data Logger’s Record

4.5.6.2 LVC Simulation Test Criteria (Pass/Fail Sheet)

Although the overall assessment of the LVC simulation is passed, we identified many of the problems that need to be addressed. The found problems are dealt with in Section 4.6. Table 11 summarizes the LVC simulation test criteria.

Table 11: LVC Simulation Test Criteria

Number	Criteria (Requirement)	Pass/Fail
1	• Successful representation of the F-16 flight detection result by Air Defense Radar in AddSIM through Data Logger.	Pass
2	• Successful providing of calculated engagement result (evasion or hit) from SA-8 SAM's missile attack.	Pass
3	• Successful representation of hit (crash of flight) through Data Logger.	Pass
4	• Target federation's situation is displayed on the Web browser.	Pass

4.6 Phase 4: Case Study Findings

This section describes the findings identified from the case study results. We evaluated and analyzed the case study's findings and then identified the problems (or limitations). Although, the overall interoperability assessment on the LVC simulation case study was successful, adjustments of many environment variables to resolve problems between SIMbox, VR-Forces and AddSIM were required. Contributing problems of the case study's results are listed in Table 12.

Table 12: Problems from LVC simulation case study results

Problems	Descriptions
Problem 1	Lack of Interaction between Simulation Entities
Problem 2	Lack of Reusability
Problem 3	Lack of Scalability and Interoperability of HLA Federation
Problem 4	Limited Capability of CGFs (or SAFs)
Problem 5	Limited Reference Models in Database
Problem 6	Limited Correlated Terrain Databases (TDBs) Representation
Problem 7	Limited Use of the Simulation Systems for Multipurpose
Problem 8	Limited Analysis of Engagement Result

The following subsections describe each contributing problem respectively.

4.6.1 Problem 1: Lack of Interactions between Simulation Entities

Entity is defined as "any distinct person, place, thing, event or concept where information is maintained or something which exists as a particular and discrete unit" (SISO, 2007). Interaction is an attempt to modify the state of the object by another object. For instance, an indirect fire, fuel supply and communication are all examples of interaction (Tolk, 2012). From the case study, we found the lack of interaction between entities of AddSIM, SIMbox and VR-Forces. In the case study, before configuring simulation environment, SA-8 SAM launched the missile to F-16 flight, but F-16 flight was not destroyed.

In order to resolve this problem, the case study framework needed adequate simulation entity mappings to achieve proper interoperability and required interaction in the defined Air Defense Engagement scenarios.

In particular, the HLA entities' definition and interactions handling is done through a *DisEntitiesMap.*XML file containing both generic and specific translations. Figure 66 depicts part of default XML entities mapping scheme provided by the SIMbox simulation system. New XML files with generic and specific entities mapping schemes can be created to implement the HLA compliance of all acting Live, Virtual and Constructive simulation systems and their corresponding scenarios in a distributed simulation environment.

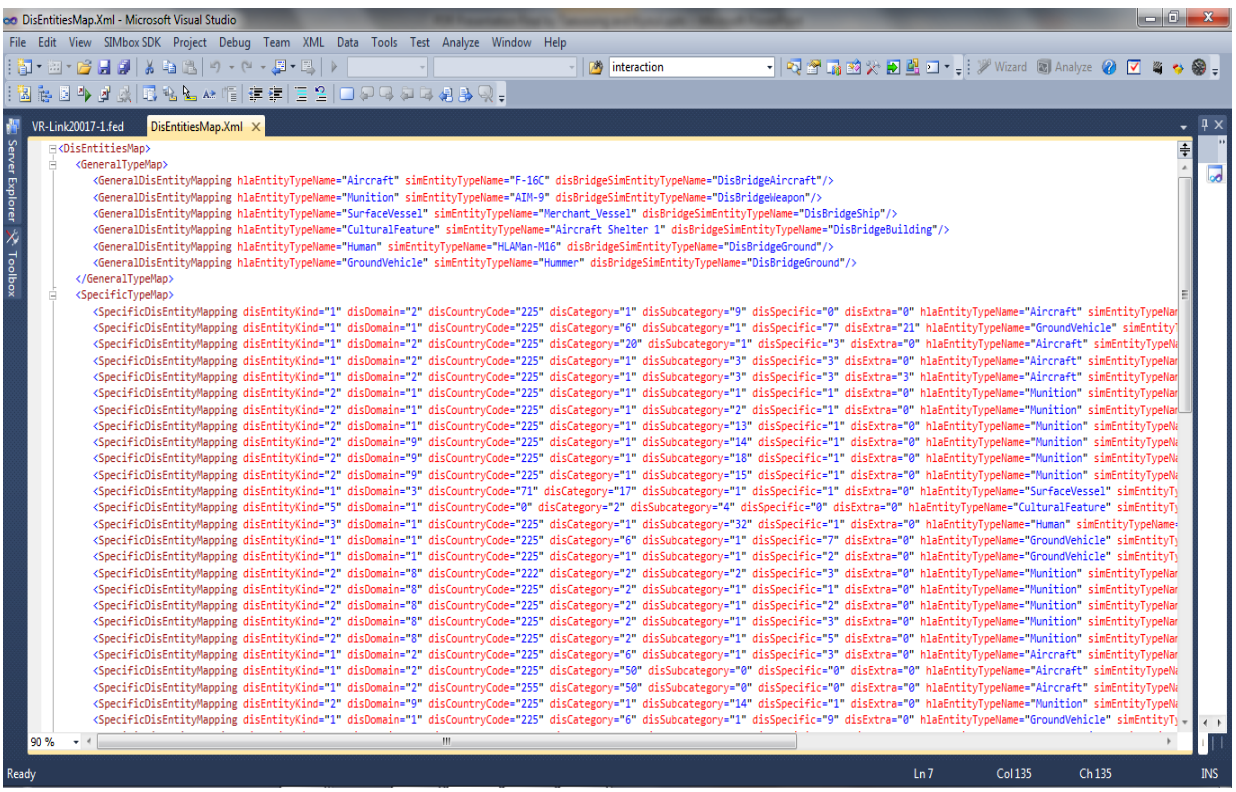


Figure 66: Entities Mapping in DisEntitiesMap.Xml

In addition to the mapping problem of the simulation entities above, the mapping of simulation attributes and simulation events within the SIMbox simulation system has a particular way to handle *Weapon Loadout Data*. In the HLA based federation, the creation and removal of weapon entities, and their data handling and translation mechanism are implemented similar to the DIS entity mapping required for the SIMbox HLA Entities. The loadout properties defined in the scenario definitions have to be mapped to an XML file called *LoadoutAuxiliaryData.xml* in the SIMbox HLA content extension implementation. The weapon *Loadout Auxiliary Data* is required for proper interoperability between simulation systems. The required HLA entity data mappings were implemented and adequate interoperation and desired level of interaction between simulation systems (or federates) were accomplished in the Air Defense Engagement scenarios.

4.6.2 Problem 2: Lack of Reusability

Inconsistency of the object models is a major cause of interoperability problems. Each entity model of AddSIM, SIMbox and VR-Forces has his own characteristics. Therefore, we examined the object model used in each simulation system.

4.6.2.1 AddSIM's Model

Modeling framework in AddSIM has been developed upon Base Object Model (BOM), SISO standard for simulation object model since 2006. SISO developed BOM to enable composability and reuse for HLA simulation. Therefore, BOM standard provides a general purpose about object modeling architecture for defining components to be represented within an LVC simulation environment. In addition, BOMs may well be used to characterize the combat models, including the predicted behavior of interacting systems, individuals, and other entities. Figure 67 shows the BOM's structure (Tolk, 2012).

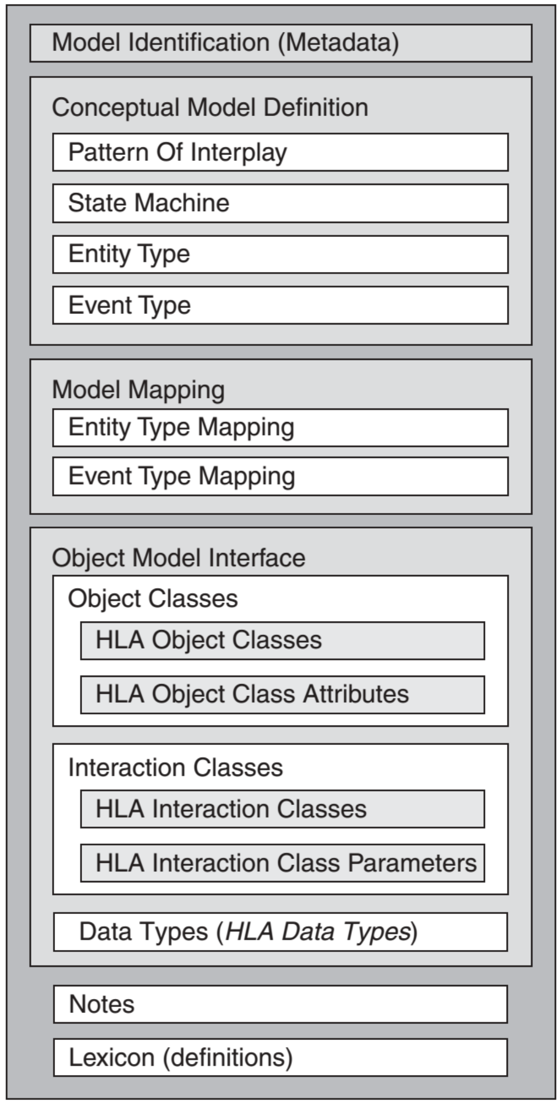


Figure 67: BOM’s Structure

Source: Tolk (2012)

4.6.2.2 SIMbox's Model

The flight simulator was developed by the SIMbox Software Development Kit (SDK). In the SDK provides three object component types: The Logic Object Component (LOC), the Console Object Component (COC) and the Output Object Component (OOC) which are basic system components of all simulation entities in the SIMbox. LOC is responsible for an entity's behavior such as steering and motion. COC is responsible for an entity's internal display. OOC is responsible for entity's external output. Table 13 summarizes the definitions and the responsibilities of each object component type.

Table 13: Three Object Component Types in SIMbox

Type	Definition/Responsibility
Logic Object Component (LOC)	• Logical state of the system • Entity's behavior • Exposing the state as attributes (Token) • Responding to action calls • Initializing properties • For example, a fuel system LOC might expose a fuel level attribute that decreases over time
Output Object Component (OOC)	• Entity's external output (show after burner, move gears, play sounds) • External visual elements, such as external subparts • Managing the control of entity sounds • For example, a fuel warning sound will play when the fuel-low attribute is set to true
Console Object Component (COC)	• Entity's internal display (speed indicator, altitude, fuel indicator) • Rendering visual elements inside the console and to reflect the system state as a response to attribute change callbacks • For example, a fuel gauge will respond to the fuel level attribute change and reposition the gauge needle

The entity object components are integrated and implemented by the SIMbox simulation engine. Figure 68 shows the partial LOCs and COCs of F-16 flight.

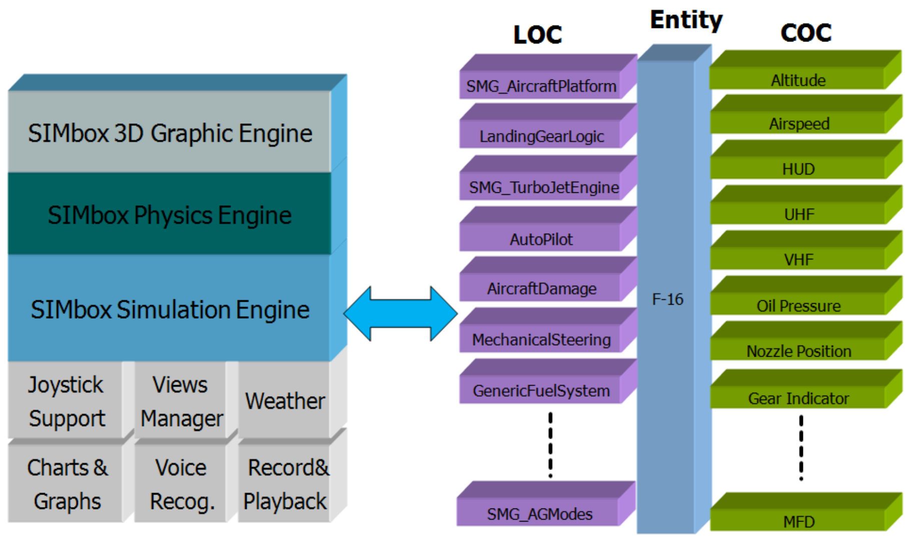


Figure 68: LOCs and COCs of F-16 flight

The SA-8 Surface to Air Missile (SAM) is low-altitude, short-range tactical SAM system. Figure 69 shows the partial LOCs and COCs of SAM.

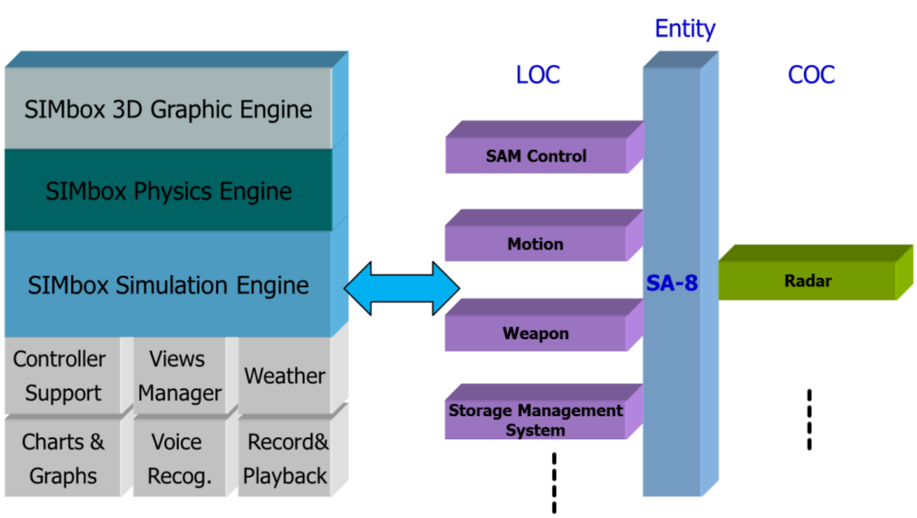


Figure 69: LOCs and COCs of SA-8 SAM

4.6.2.3 VR-Forces's Model

We describe the basic structure of VR-Forces's entity. VR-Forces does not use derived classes to distinguish different types of entities, such as ground vehicles, missiles, and so on.

An entity is expected to have the following subcomponents:

- State repository
- Network interface
- Task Manager
- Plan Manager
- Component Manager

4.6.3 Problem 3: Lack of Scalability and Interoperability of HLA federation

In the case study, each simulation system interoperated as a part of a simulation environment. However, each simulation system had limited capability to support interoperability and scalability.

4.6.3.1 Lack of Scalability

The scalability issue occurs when a large number of entities have been created in the simulation scenario. According to the definition of DoD M&S glossary, scalability is that "the ability of a distributed simulation to maintain time and spatial consistency as the number of entities and accompanying interactions increase" (DoD, 2011).

In any of the simulation tests of SIMbox, we identified stopping phenomenon we made a number of entities. This stopping phenomenon is can be highlighted as a big problem in the real-time simulation.

The number of the entity that is created depends on the purpose and scale of the scenarios. The Performance problem occurs when the scenarios fail to fully disperse the workload or event operations associated with any entities becomes a bottleneck. One way to alleviate this problem may be to use a variety of embedded grid-computing techniques to parallelize the processing of a single event (Jeffrey S. Steinman, 2013). This technique is described in detail in Section 4.7.6.

4.6.3.2 Lack of Interoperability of HLA Federation

The High Level Architecture (HLA) is the most commonly used SSA from the M&S community. The HLA enables reuse and interoperability of simulation systems through defining a template for object models that can be used to exchange data, setting rules for simulation system and applications, and standardizing communication interface between simulation applications and simulation infrastructure (Çelik, Gökdoğan, Öztürk, & Sarikaya, 2012).

Since HLA federations are composed of over two kinds of the loosely coupled simulation systems (called federates), it can be thought of as “enterprises," each of which may be considered to provide the ability to operate the different functions in their time scales. Enterprises mean that it integrates multiple disjointed applications in loosely coupled distributed simulation systems. Enterprises (or federations) consists of several simulation systems (or federates) that may run internally on one or more local machines. Enterprises can be locally or geographically distributed across arbitrary networks. However, in such a simulation environment, communications may be often sporadic or irregular.

In an ideal world, the combined set of simulation systems within an HLA federation spans the required performance of the simulated system for its intended purpose. However, loosely coupled federations may face a conceptual modeling problem, making it very difficult to prove the simulation results. Federates (or simulation systems) within a federation (or simulation environment) often have duplicate models, that further aggravate the problem of validation. The case study showed it to be very costly to integrate federates into multiple federations because we used a universal bridging tool. Such a problem is further generated especially when object models are different, startup procedures are specific to each federation, tools are federation-specific, scenario descriptions have different formats, etc. Therefore, run-time performance of HLA federations may be far from ideal as we expected (Jeffrey S. Steinman, 2013).

We need to configure the HLA federate environment when using HLA/RTI.

First, an RTI must be installed on each computer that is running an HLA federate. Federates must be able to find the RTI libraries (*.dll* or *.so*.). User should accomplish this by adding the path to the RTI's lib directory to the path environment variable for user operating system.

Second, all federates in a federation must use the same manufacture RTI such as MÄ K RTI or Pitch pRTI, configured in the same way, use the same FED file (FDD file in HLA 1516), and each federate must be able to find the FED file.

Third, the most important issue for compatibility when running applications using the HLA is to ensure that each federate is using same version of the RTI and the same FED file. In the case study, there was no common FOM representation to use (or FED file) between VR-Forces, SIMbox and AddSIM. Namely, a set of federates must agree on a common FOM in order to communicate. Therefore, too much time for configuring was required to make it become interoperable between them.

Finally, all federates in a federation must use the same version HLA such as HLA 1.3, HLA 1516 or HLA 1516 evolved. The consistency in FOM format is necessary for the interoperability. The main reason is that HLA 1.3 and HLA 1516 use different names for the Root classes of the Object and Interaction class hierarchies. A 1.3-style FED file requires a Root class called *ObjectRoot*, whereas a 1516-style XML files requires a Root class called *HLAObjectRoot*. If the Logger is playing back an HLA-1.3-based Logger file into a federation that is using a 1516-based XML file, it might come across an instance of a class called, for example, *ObjectRoot.Vehicle*. If it tries to register an object of this class, the RTI will complain that no such class exists. There might be a class called *HLAObjectRoot.Vehicle* in the current FOM, but the RTI does not know that this is actually the same class. Therefore, both RTI and federates will not realize that these classes were intended to be same. The subscribing federate will also fail to discover any objects that the publishing federate registers (MÄ K).

The following subsections cover the interoperability capability of each simulation system.

4.6.3.2.1 AddSIM's Interface

AddSIM provides three external interfaces such as C/C++, Matlab, HLA/RTI and DIS interface as shown in Figure 70 (Lee et al., 2012). In terms of HLA/RTI interface, AddSIM was designed as federates for the joining to HLA-based simulation environment that are called “federation” in HLA. AddSIM is compliant with HLA 1516 which is a SISO Dynamic Link Compatibility (DLC) version of HLA 1516-2000, and HLA Evolved is HLA 1516-2010 with the exception of the HLA 1.3 specification. It can be a great disadvantage because the HLA 1.3 version is more commonly used than HLA 1516.

AddSIM also uses DIS to support interoperation with other simulation systems. Using HLA, AddSIM was included as a federate in the case study.

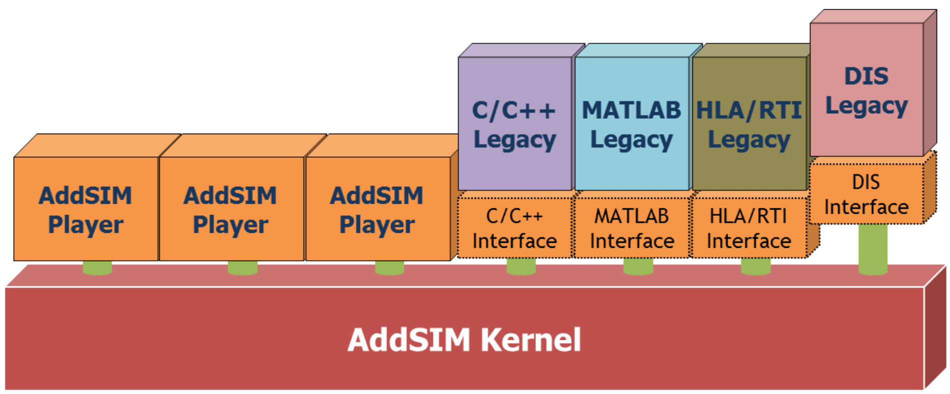


Figure 70: External Interface of AddSIM

<u>4.6.3.2.2 VR-Forces’s Interface</u>

VR-Forces is compatible with both the DIS and HLA simulation standards. VR-Forces also supports the use of HLA Data Distribution Management (DDM) as a means of managing large numbers of entities dispersed over wide areas. VR-Forces supports both the HLA 1.3 specification and the SISO DLC version of the IEEE 1516 specification. VR-Forces has built-in support for the HLA RPR-FOM and can support other FOMs through the FOM Mapping feature. For either HLA specification, user can run simulation using one of several different versions of the RPR FOM. VR-Forces supports time management for HLA exercises. A simulation connection of VR-Forces specifies the connection parameters for a DIS or HLA simulation connection as shown in Figure 71.

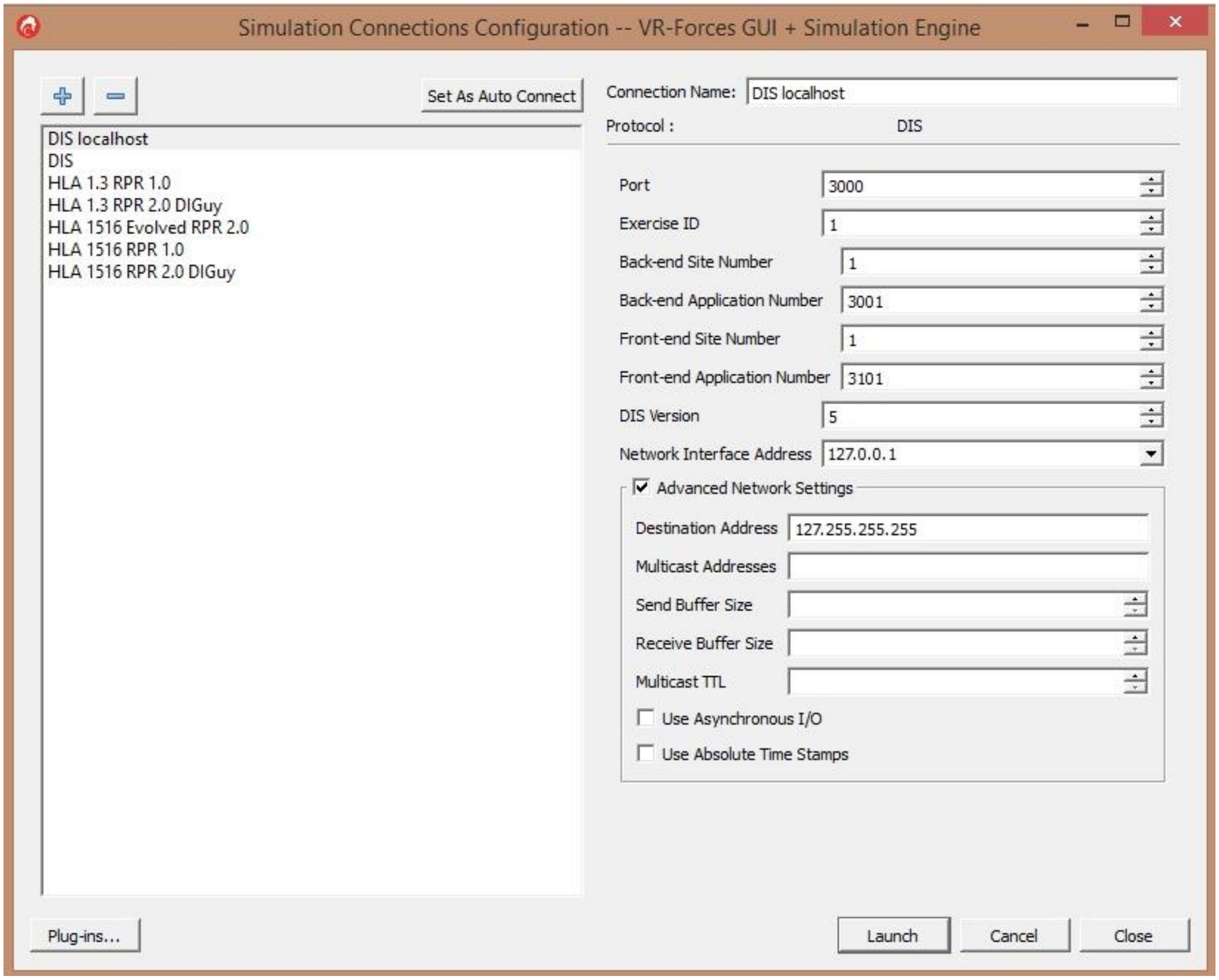


Figure 71: Simulation Connection Configuration of VR-Forces

VR-Forces comes with the following connection configurations: DIS (port 3000), HLA 1.3 RPR FOM 1.0, HLA 1.3 RPR FOM 2.0, HLA 1516 RPR FOM 1.0, and HLA 1516 RPR FOM 2.0

4.6.3.2.3 SIMbox's interface

SIMbox is HLA compliant and FOM agile, enabling integration with external components. Figure 72 shows SIMbox HLA extension.

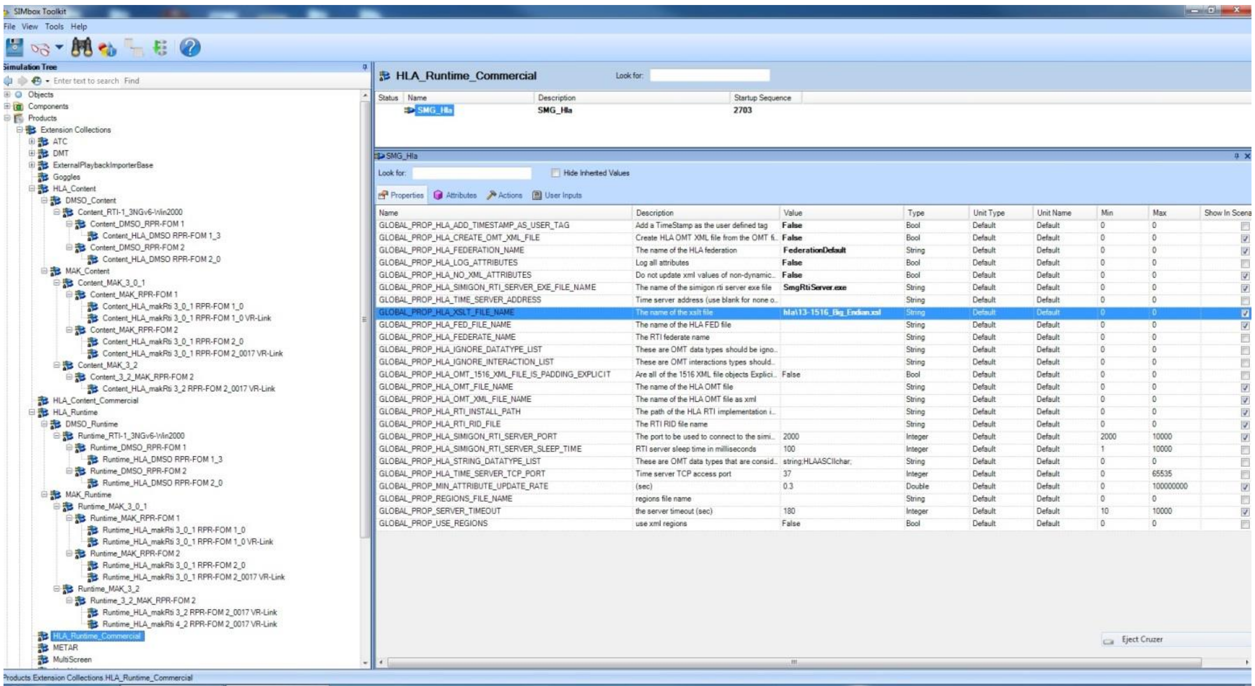


Figure 72: SIMbox HLA extension

4.6.4 Problem 4: Limited Capability of Computer Generated Forces (CGFs) (or Semi-Automated Forces (SAFs))

Computer Generated Forces (CGF) means some simulation entities which are created and controlled by the computer in the battlefield simulation environment. CGF also sometimes referred to as Semi-Automated Forces (SAF) is a very important component in Constructive simulation system and is increasingly being used to control multiple entities in Synthetic Environments (SEs).

According to the U.S. DoD Modelling and Simulation (M&S) Master Plan, the definition of CGF is as follow:

"A generic term used to refer to computer representations of entities in simulations which attempts to model human behavior sufficiently so that the forces will take some actions automatically (without requiring man in-the-loop interaction)"(DoD, 1995).

From the case study, we compared the ability of the CGF between AddSIM, SIMbox and VR-Forces. Then, we identified what needed to be improved.

AddSIM does not yet include any models of a human decision maker. SIMbox showed some ability of CGF between F-16 flight and SA-8 SAM. Among them, VR-Forces showed the most powerful and flexible CGF. VR-Forces provides both a set of APIs for creating CGF applications, and an implementation of those APIs. The simulation API gives the developer or user control over: behaviors, components, entity types, parameters, messages, resources, tactical graphics, plans and tasks.

In addition, One Semi-Automated Forces (OneSAF) is one of the well-known CGF simulation systems in the U.S. Army. OneSAF provides individual battlefield CGF such as tanks, helicopters and soldiers. OneSAF also supports aggregate units, to the Brigade level. User can operate in ether a fully automated mode or under the control of the human operator via their organic command and control systems or role players using an OneSAF GUI.

In conclusion, to improve limitations of current CGF (or SAF) in AddSIM, obviously, the more realistic CGF with AI is required.

4.6.5 Problem 5: Limited Reference Models in Database

We compared the AddSIM with several Commercial-Off-The-Shelf (COTS) simulation systems in the market like MÄ K's VR-Forces and SimiGon's SIMbox simulation systems that have the goal of providing a tactical environment in terms of the reference model.

VR-Forces is the CGF simulation system for a wide variety of battlefield entities and weapon systems. On the other hand, the number of reference models in SIMbox and AddSIM simulation systems was relatively small comparing it to VR-Forces.

4.6.6 Problem 6: Limited Correlated Terrain Databases (TDBs) Representation

This section describes the limited correlated terrain databases between AddSIM, VR-Forces and SIMbox from the case study.

SIMbox uses the industry-standard OpenFlight terrain format, and VR-Forces also supports the OpenFlight terrain format. VR-Forces was needed to display the same Las Vegas terrain of SIMbox to be interoperable. We loaded the SIMbox's Las Vegas terrain format into VR-Forces successfully as shown in Figure 73. However, AddSIM does not have the terrain application to support load from another terrain format.

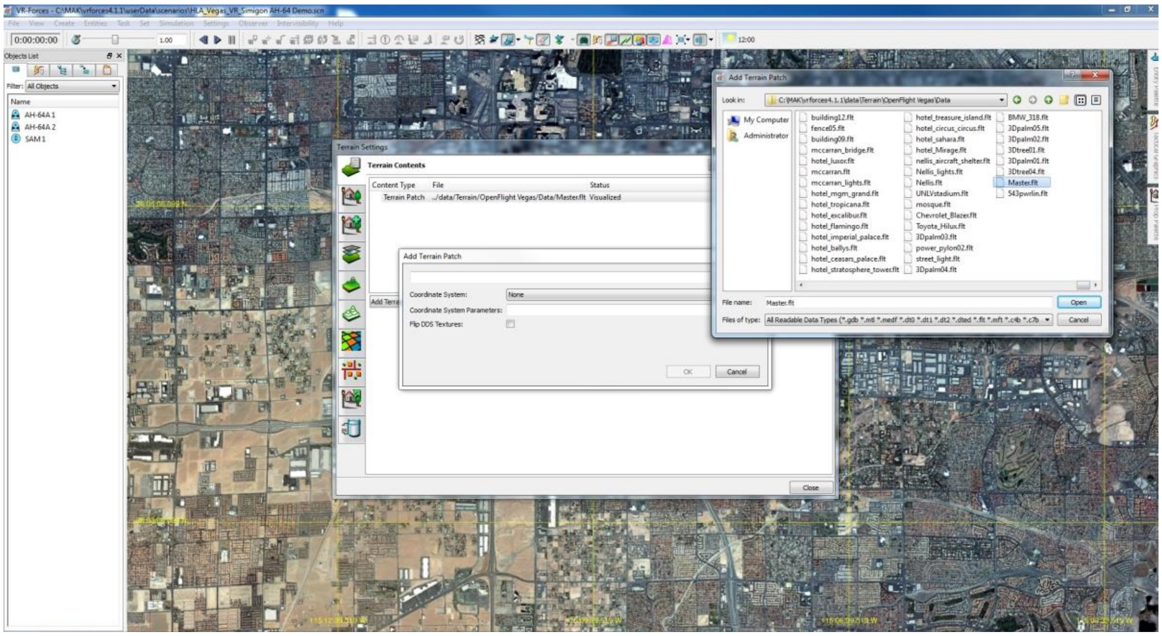


Figure 73: Loading SIMbox's LasVegas terrain format into VR-Forces

In addition, we found some problems with the coordinate between AddSIM and VR-Forces. VR-Forces includes several databases that use Universal Transverse Mercator (UTM) coordinates. On the other hand, AddSIM uses the Latitude/Longitude (decimal radians) coordinate system. Therefore, because of the differences in terrain databases among exercise participants, entities can sometimes appear to be underground or hovering above the terrain surface. VR-Forces supports the following coordinate systems as summarized in Table 14, but coordinate issue occurred, and a lot of work has been required to resolve the issue. The coordinate problems must be resolved because the issue is associated with a target acquisition, entity movement and fair fighting.

Most simulation systems are not common in the distributed simulation systems to share a single representation of the synthetic environment over the network. Mostly, each of the simulation systems has its own internal representation of the synthetic environment. Often, there are different methods defining the bare earth terrain relief and extracting and representing features in the area of interest. In addition, there are differing terrain database formats and tools underlying simulation applications. Polymorphism differences in the representation and different target formats of TDB result in correlation problems between multiple simulation systems. The correlation problems include numerical inaccuracy, algorithmic, parametric, a temporal inconsistency.

Table 14: Coordinate systems

Coordinate System	Description
Universal Transverse Mercator (UTM)	Location is displayed as two position fields and one altitude field. Position coordinates are displayed in the Universal Transverse Mercator system. The first position field displays the zone and the x location in meters. The second position field displays the y position in meters.
Geocentric	Location is displayed as three position fields. Position coordinates are displayed in the geocentric coordinate system. The position fields are always in meters.
Military Grid Reference System (MGRS)	Location is displayed as two position fields and one altitude field. Position coordinates are displayed in the Military Grid Reference System. The first position field displays the zone. The second position field displays the grid location. The precision controls the number of digits used in the grid display.
Latitude/Longitude	Location is displayed as two position fields and one altitude field. Displays coordinates in the latitude and longitude using the geodetic WGS84 coordinate system. Each angle will be displayed in degrees : minutes : seconds with seconds displaying base 10 fractional seconds.
Latitude/Longitude (decimal radians)	Location is displayed as two position fields and one altitude field. Position coordinates in the latitude and longitude using the geodetic WGS84 coordinate system. Each angle will be displayed in decimal radians
Database	Location is displayed as three position fields. Location is displayed using VR-Forces's current internal Cartesian database system. The position fields will be displayed using the current distance units.

4.6.6.1 Terrain Database of AddSIM

AddSIM provides the post-analysis module to analyze the simulation result and visualization module using SIMDIS to play back the entire simulation (Lee et al., 2012). The post-analysis module that is simulation output formats consists of CSV file, analytic report, and visualization format (SIMDIS format). Figure 74 shows a snap shot of an anti-air missile engagement using SIMDIS. SIMDIS which is developed by a Naval Research Laboratory is a set of software tools. It provides 2D or 3D interactive video display and graphics of live and post processed simulation, operational data and test (U.S._Naval_Research_Laboratory). However, AddSIM does not have the detailed terrain database to be compatible with VR-Forces and SIMbox as show in Figure 75. Therefore, we cannot observe the movement of entities through the AddSIM's terrain GUI during simulation execution.

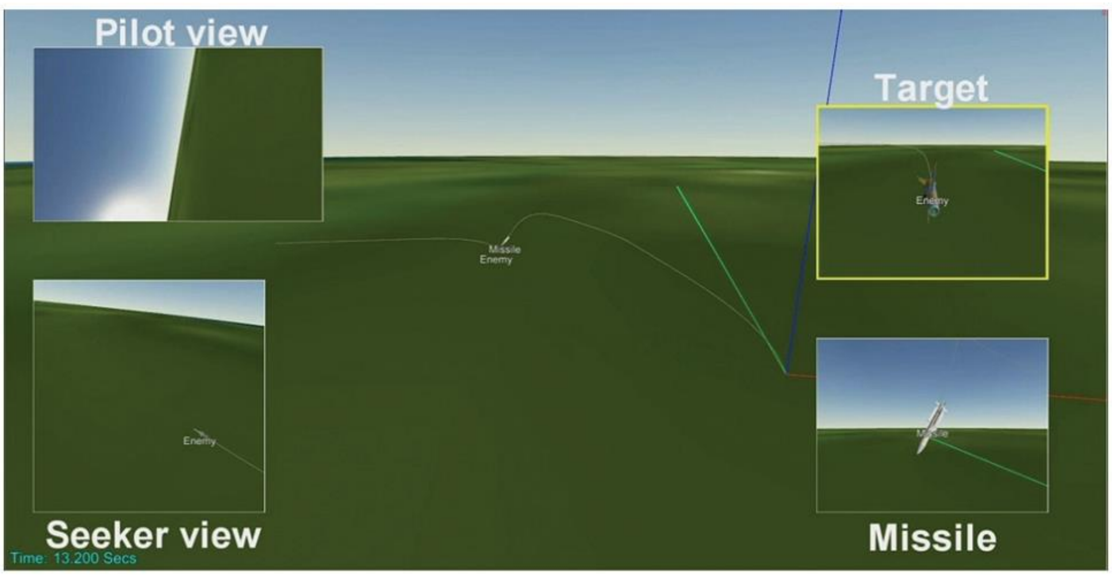


Figure 74: Screen Shot of SIMDIS

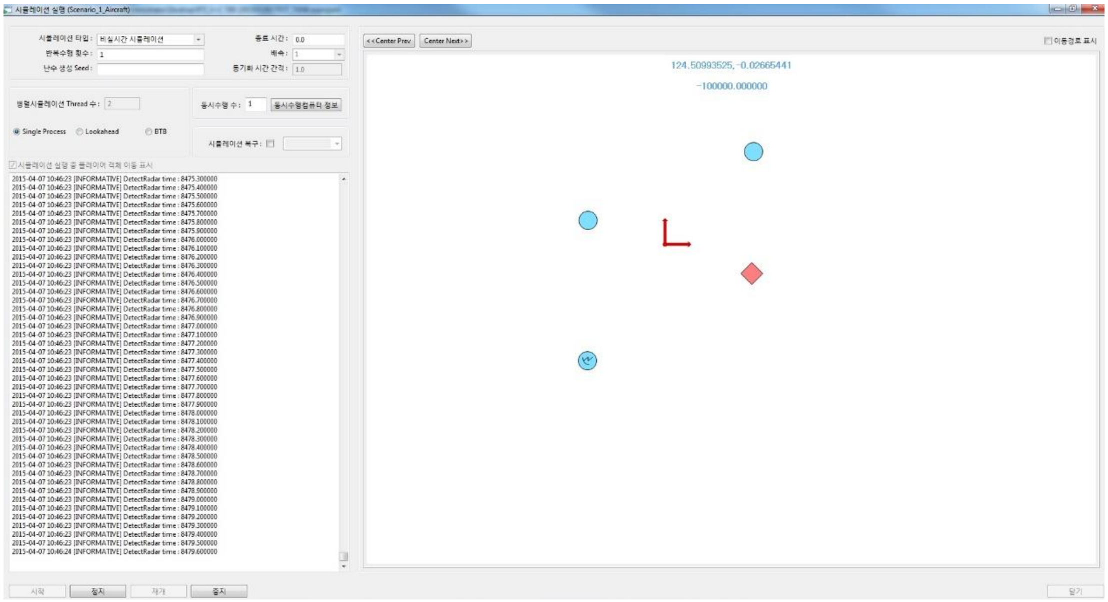


Figure 75: AddSIM Terrain

4.6.6.2 Terrain Database of VR-Forces

MÄ K VR-Forces's terrain database is one where GDB is a collection of polygons that have associated with attribution such as soil type. VR-Forces allows user to build user's terrain at runtime using a variety of databases and vector formats. The MÄ K Terrain Database Tool (TDB Tool) allows users to create GDB terrains for use with VR-Forces, and import vector data. VR-Forces supports the following database formats (MÄ K, 2011):

- The UTM projection and the Lambert conical conformal projection in CTDB C4B, C7B, and C7L databases
- OpenFlight UTM and flat earth databases
- MÄ K Terrain Format (GDB)
- Digital terrain elevation data (DTED) databases

- Shape files
- Flat earth
- VMAP
- DFAD and DFD files

Figure 76 shows the VR-Forces terrain database.

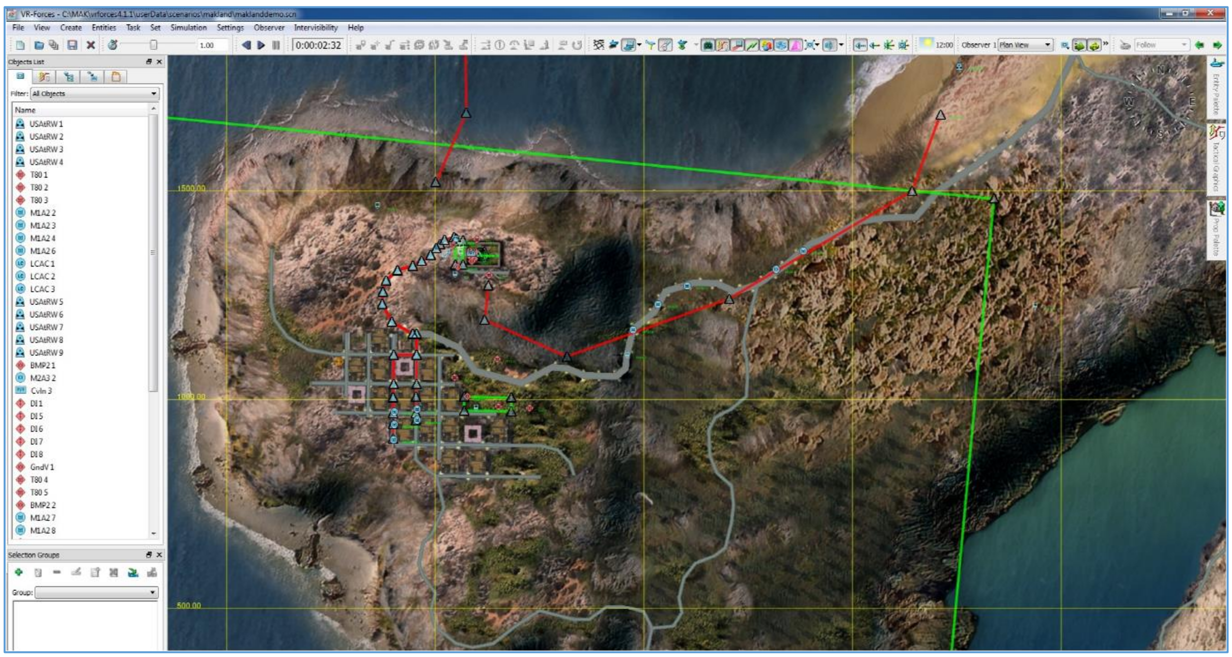


Figure 76: VR-Forces terrain database

4.6.6.3 Terrain Database of SIMbox

SIMbox uses the industry-standard OpenFlight terrain format. Terrain databases can be created from standard geodata images, map and elevation data, as well as GIS data such as roads, Vertical Obstruction data, building outlines and more (SimiGon). However, SIMbox does not use Global World Terrain to reduce the overall size of the installation. A separate Global World

Terrain is available to users on request, and this terrain can be installed separately. In addition, the graphic engine supports UTM projection in terrains (SimiGon). The Figure 77 shows SIMbox’s Las Vegas OpenFlight terrain format.

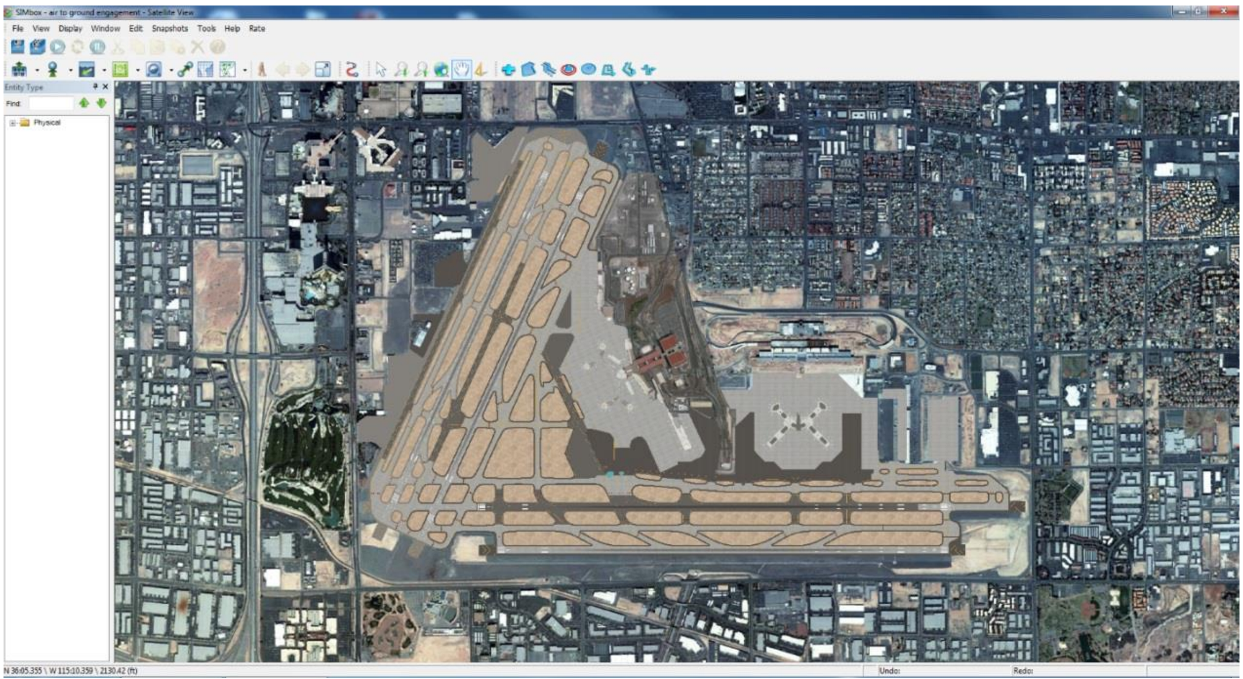


Figure 77: SIMbox terrain database

4.6.7 Problem 7: Limited Use of the Simulation Systems for Multipurpose

Until now, most simulation systems were developed to achieve one goal among research & development (R&D), analysis, training exercise, military operation or acquisition. We identified that each simulation system of the case study has its own goal.

4.6.7.1 Use of AddSIM

AddSIM simulation system is an engineering and engagement level for the weapon systems R&D by ADD and defense industries. Sample model in AddSIM is developed as an engineering level simulation of a specific weapon system, which is used to analyze the Measure of Performance (MOP).

4.6.7.2 Use of VR-Forces

VR-Forces is typically used for training at the tactical level in order to provide a very high level of detail of the battlefield. Therefore, VR-Forces is a simulation system that is limited to the analysis and experimental purposes.

4.6.7.3 Use of SIMbox

SIMbox simulation system is also a high-fidelity 3D training simulation system. Accordingly, SIMbox simulation system is limited as a tool for analysis and experiment as well.

4.6.8 Problem 8: Limited Analysis of Engagement Result

In the Air-Defense Engagement scenario, we were unable to get the detailed information about the engagement result from AddSIM, SIMbox and VR-Force.

4.6.8.1 Battle Damage Assessment (BDA) of AddSIM

Since the current AddSIM is in the development process, it does not provide detailed information about the engagement results. In AddSIM, when the player was destroyed, simulation was automatically shut down.

4.6.8.2 Battle Damage Assessment (BDA) of VR-Forces

The damage value of the entity can be checked in the Entity Information dialog box as shown in Figure 78. Figure 78 shows no damage to the F-16 flight.

Figure 78: Damage Value in VR-Forces

An entity should look different if it is damaged or destroyed. Table 15 summarizes the damaged appearance of an entity.

Table 15: DIS Damage Appearance

Damaged appearance of an entity	0	None
	1	Slight
	2	Moderate
	3	Destroyed

4.6.8.3 Battle Damage Assessment (BDA) of SIMbox

In SIMbox, we could see the damage value (0~100) of the entity. For example, damage value 0 is indicative that the entity has not received any attack damage, while damage value 100 means that the entity is destroyed. In the Air-Defense Engagement scenario, because the plane crashed to the ground when shot down by a missile and destroyed, the damage value was always 100. Therefore, we need to further analyze the battle damage mechanism of different entities except for the flight entity in SIMbox. The following sections describe the test of engagement between the F-16 flight and the T-72 Tank.

<u>4.6.8.3.1 Engagement Scenario between F-16 Flight and T-72 Tank</u>

This scenario consists of ten T-72 tanks (Tank Company), one SA-8, two Mig-29s and four F-16Cs. The main goal of the operation is for two F-16Cs to destroy ten T-72 tanks. Another goal is for two F-16Cs to engage two Mig-29s that are circling at an altitude to protect ten T-72 tanks. The SA-8 is also located to protect the T-72 Tank Company from the F-16Cs attack. The situation map and each entity are shown in Figure 79.

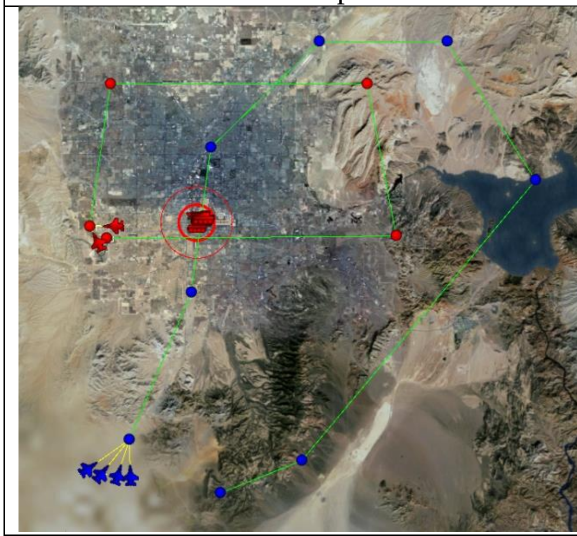
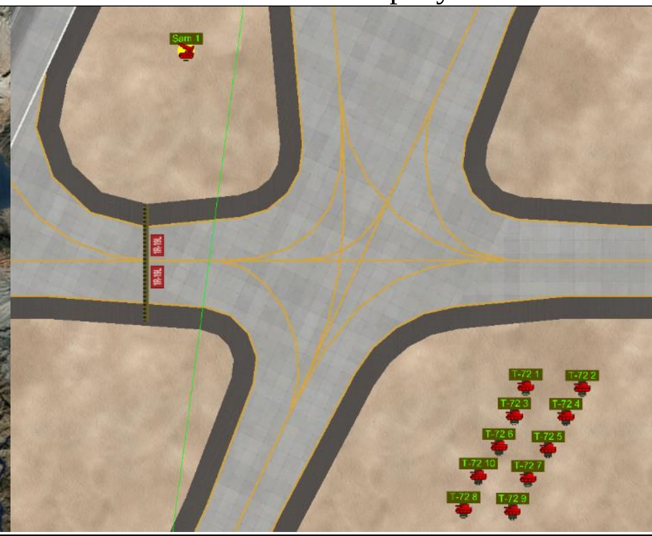
Situation Map in 2D	T-72 Tank Company in 2D
	
T-72 Tank Company in 3D	SA-8 SAM in 3D
	
F-16 Flight in 3D	Mig-29 Flight in 3D
	

Figure 79: Situation Map and F-16 Flight, Mig-29 Flight, T-72 Tank and SA-8 SAM

4.6.8.3.2 Engagement Result between F-16 Flight and T-72 Tank

When the F-16C attacks T-72 Tank Company, the air to surface missile from the F-16C hits the T-72-4 and the explosion point that is a red dot is shown in Figure 80. As a result, T-72-4 was eliminated, and T-72-3 and T-72-2 received damages form the explosion and showed heavy smoke and light smoke respectively.

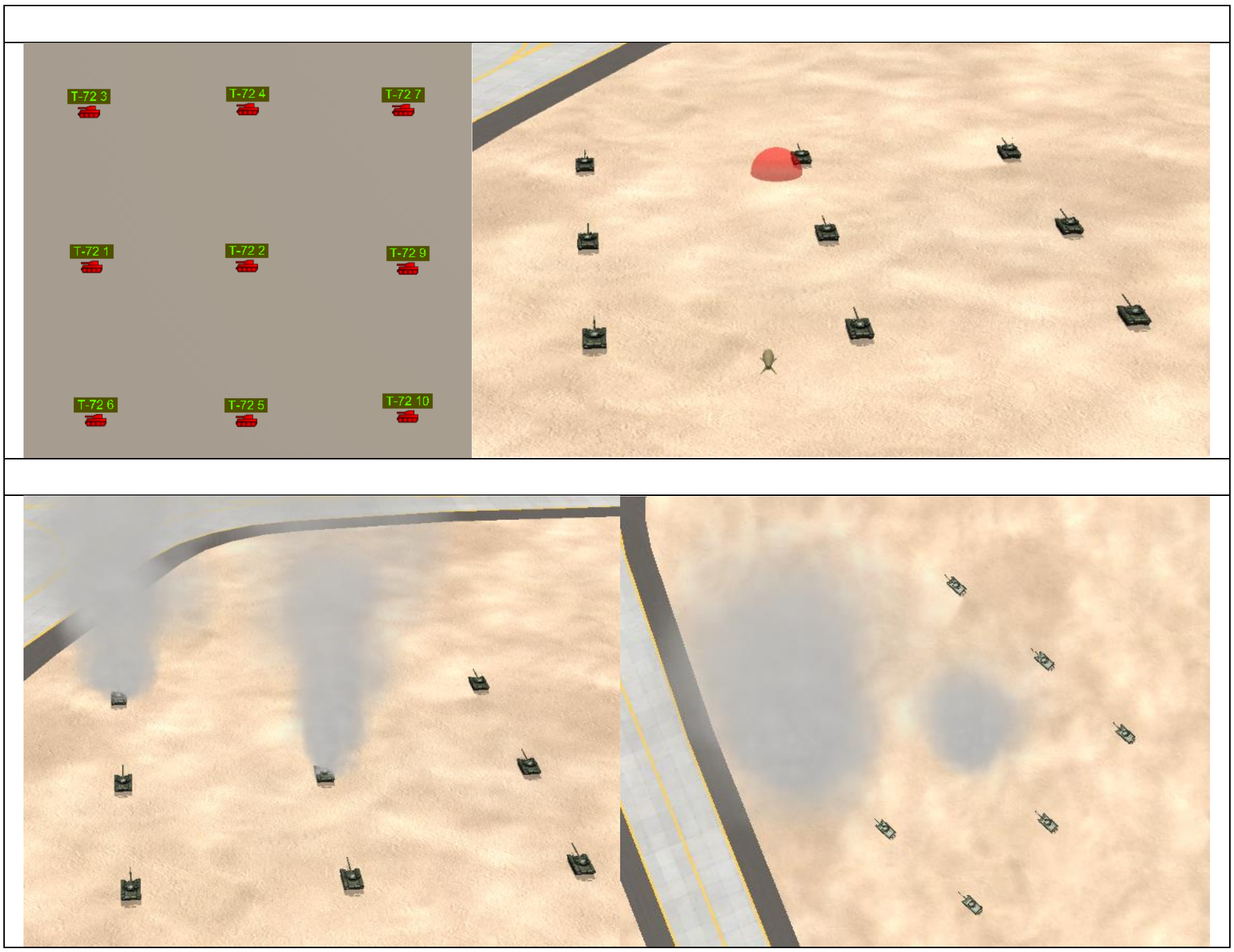
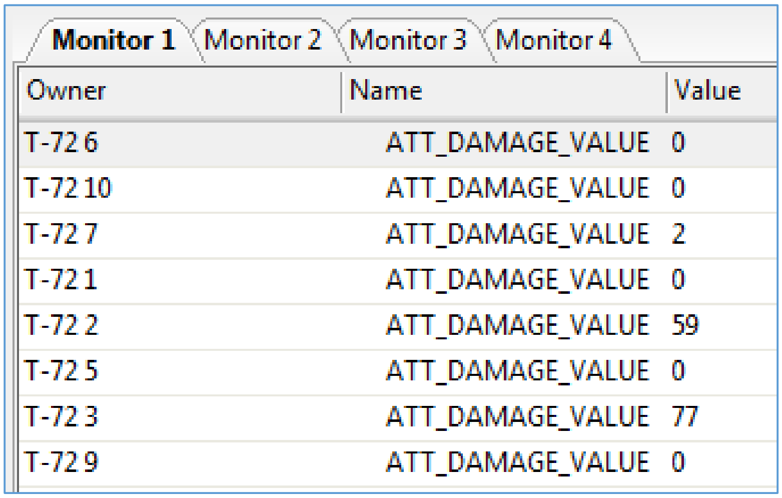


Figure 80: F-16C’s attack to T-72 Tank Company

The damage value of these tanks is shown in Figure 81. T-72-4 tank was eliminated because its damage value is 100. T-72-3, T-72-2 and T-72-7's damage value is 77, 59 and 2 respectively.



Monitor 1 | Monitor 2 | Monitor 3 | Monitor 4

Owner	Name	Value
T-72 6	ATT_DAMAGE_VALUE	0
T-72 10	ATT_DAMAGE_VALUE	0
T-72 7	ATT_DAMAGE_VALUE	2
T-72 1	ATT_DAMAGE_VALUE	0
T-72 2	ATT_DAMAGE_VALUE	59
T-72 5	ATT_DAMAGE_VALUE	0
T-72 3	ATT_DAMAGE_VALUE	77
T-72 9	ATT_DAMAGE_VALUE	0

Figure 81: Damage Value of T-72 Tanks

The damage value is proportional to the proximity of explosion point as shown in Figure 82. T-72-3 with much damaged is 139.00 feet away from T-72-4. T-72-2 is 135.29 feet away from T-72-4. T-72-7 with less damage, is 166.33 feet away from T-72-4.

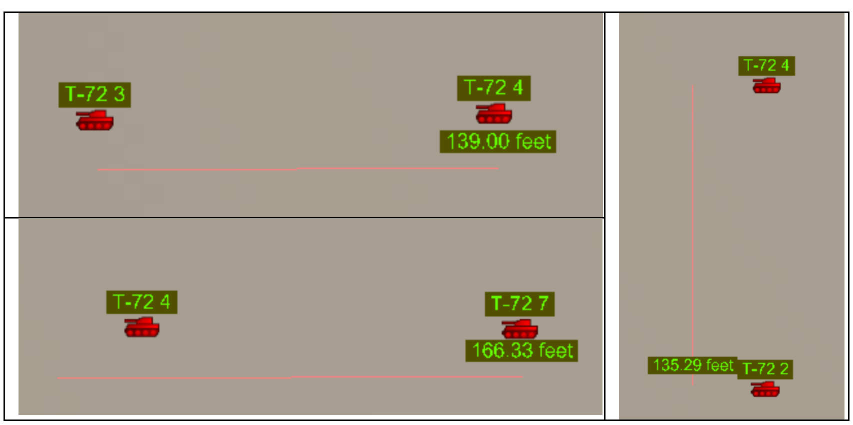


Figure 82: The Damage Value according to the Distance of an Explosion

4.7 Phase 5: Case Study Lessons Learned

This section discusses the lessons learned from the case study. We learned a great deal about the LVC simulation and about what technologies work best under current LVC simulation circumstances. Table 16 summarizes an overview list of lessons learned that must be considered with the goals for constructing the roadmap for LVC-ITA.

Table 16: List of Lessons Learned

From Problem No.	Lesson Learned No.	Description
Problem 1 Problem 2	Lesson Learned 1	• Need for a Common Standard Simulation Entity
Problem 4 Problem 5	Lesson Learned 2	• Need for an Entity Level Simulation Systems
Problem 1 Problem 2	Lesson Learned 3	• Need for Common Standard-Defense Conceptual Modeling
Problem 4	Lesson Learned 4	• Need for Computer Generated Forces (CGF) (or Semi-Automated Forces (SAFs))
Problem 3	Lesson Learned 5	• Need for Multiple SSAs Compliancy
Problem 3	Lesson Learned 6	• Need for Scalability Capability of Simulation Systems
Problem 6	Lesson Learned 7	• Need for Common Correlated Terrain Databases (TDBs)
Problem 3	Lesson Learned 8	• Need for a New Common Standard Simulation Architecture (C-SSA)
Problem 2 Problem 3	Lesson Learned 9	• Need for a Product Line Architecture Framework (PLAF) Concept
Problem 7	Lesson Learned 10	• Need for a Simulation System to Support Multiple M&S Applications and LVC simulations
Problem 8	Lesson Learned 11	• Need for a Battle Damage Assessment (BDA) Application
Problem 3	Lesson Learned 12	• Need for a General Bridging Tool

4.7.1 Lesson Learned 1: Need for a Common Standard Simulation Entity

Lesson learned 1 is the need for a common standard simulation entity. It was derived from problem 1 and is the lack of interactions between simulation entities, and problem 2 is the lack of reusability.

In this section, I would like to emphasize the need for a common entity. Prior to the describing the common entity, we need to clearly distinguish the differences between entity, unit, and object.

Unit is organized as a military organization, such as platoon and company. They have a certain scale and are composed of a variety of subordinate units such as specific combat. For example, a regiment is composed of three battalions. Entity is an element or individual object in a simulation system, such as a soldier and flight; that is represented in the simulation and can be broken into smaller parts. Object is a generic term used to describe the entity or unit. It has persistence and is a transient element.

Entities have complex capabilities, such as the ability to move, to take damage, to sense other entities, and to shoot munitions. Entities have a lot of information. For example, states such as speed, location, and heading, tasks such as move, patrol, follow, and fire. They can be an enemy, friendly or a neutral system.

The most demanding simulation systems are composed of many interacting entities. The entities are usually organized hierarchically such as ground entities, air entities, surface entities, life form and aggregate entities. Therefore, a variety of entities within multiple simulation systems must be able to interact with other entities at arbitrary time scale without the mutual constraints during simulation execution. This means that any entity of the simulation systems can interact and share data with any other entity at any time, and potentially regardless of how entities are dispersed through the processors, machines, and/or networks.

If there is a need to share entities between organizations using different methods for entity modeling, each organization must understand the modeling methodology of other organizations. For the interaction, a lot of time and cost will be incurred for the mapping.

Therefore, entities in a simulation system should be easy to utilize in other different simulation systems. To this end, using a single language model has to be developed for a common entity. In addition, the simulation system should provide a common entity model repository which contains continuously available entities.

In conclusion, it is necessary to develop the common entity model that ensures interoperability and reuse.

4.7.2 Lesson Learned 2: Need for an Entity Level Simulation Systems

Lesson learned 2 is the need for an entity level simulation systems. It was derived from problem 4, there is the limited capability of the CGFs (or SAFs), and problem 5, there are limited reference models in the database.

In this section, we emphasize the need of the entity level simulation system. Constructive simulation system can usually be divided into two categories on the basis of their resolution. Table 17 summarizes the classification of the constructive simulation system.

Table 17: Classification of Constructive Simulation System.

Category	Level	Objects	Terrain
High resolution	Entity	Entity, e.g. a tank, a soldier	High resolution, 200×200 km
Low resolution	Unit	Unit, e.g. a company, a battalion,	Low resolution, 4000×4000 km

Although there was no MRM problem in this case study, we would like to emphasize the necessity of entity level simulation system. We realized that it was necessary to develop an entity-level simulation with a general purpose. When several simulation systems are interconnected, there might be Multi Resolution Method (MRM) issues. Davis and Bigelow (1998) define multi-resolution modeling as follow:

- Building a single model with different levels of resolution for a problem;
- Building an integrated family of consistent models with different levels of resolution for a problem; or
- Both

Many Virtual Simulator and Live systems have already been connected by the DIS. There is a FOM called real-time platform reference (RPR) FOM based on DIS PDUs. Therefore many LVC simulation environments use the RPR FOM. Because the RPR FOM does not support entity to aggregate interactions, aggregate level simulation system is not preferred for LVC simulation (Tolk, 2012).

The following subsections describe the justification the entity-level simulation system and several models.

4.7.2.1 Entity-Level Simulation Systems.

In this section, several entity level simulation systems are described. Through these simulation systems, we can identify the features of entity level simulation systems.

Born in the 1990s, OneSAF is an entity level based simulation. OneSAF provides individual simulation objects (or entities) in battlefield.

VR-Forces supports both at the entity level and the aggregate level. VR-Forces provide functions that the user can interactively add individual entities to a simulation and aggregate them into higher echelon units.

Joint Semi-Automated Forces (JSAF) is also an entity level simulation system, which was developed in 1990. The entities can be controlled individually or as an organizational unit. JSAF is an open environment where the property, mission and behaviors of the entity can be modified.

Joint Conflict and Tactical Simulation (JCATS) also provides a large number of entities. JCATS which provides a very high level of detail such as people, activities, and buildings, supports military training and operation experimentation.

4.7.2.2 Features of Entity-Level Simulation System.

From the MRM perspective, if M&S communities are creating a simulation system based on entity-level, MRM issue does not occur, and can later easily implement LVC simulation environment.

From the cost-effectiveness perspective, the entity-level simulation system will be contributed to reduce duplicate investments in the M&S sector, improve interoperability and foster reuse across M&S assets. Furthermore, it will meet the M&S requirements of the future combat training.

From the training perspective, simulation system makes it easy to control the individual entities such as vehicles, people, and even animals. Such a system would be useful to all trainees, and supervisors because the level of resolution of entity-level simulations is more intuitive to users and directly more supportable by available test and operational data on entity performance than the relatively abstract equations of a unit-level simulation system (Tolk, 2012).

In conclusion, entity-level simulation systems shall be developed to provide a broad range of support for sea and air entities as well as for land entities.

4.7.3 Lesson Learned 3: Need for a Common Standard-Defense Conceptual Modeling

Lesson learned 3 is the need for a common standard-defense conceptual modeling. It was derived from problem 1, the lack of interaction between simulation entities, and problem 2, the lack of reusability.

This section covers the common standard-defense conceptual modeling. In distributed simulation systems, focus has been based on the ontology components in order to achieve simulation reuse and enhance interoperability.

In Section 2.1.1.2, we described the Levels of Conceptual Interoperability Model (LCIM) developed to get the theoretical basis for the interoperation between two or more simulation systems (or federates). Semantic interoperability is needed to achieve seamless interoperability between systems.

In Section 2.4, we also mentioned the conceptual model (CM) and several modeling and simulation (M&S) process related to CM such as FEDEP, SEDEP and DSEEP. The high level outputs produced by each M&S process, are deliverables of the federations (or simulation environment), reusable common components, the object model and more. Therefore, the establishment of the M&S development process is very important.

However, the assumption of DSEEP is that only one SSA will be used. The SEDEP improved the FEDEP which was usually driven by technical need and perspective (Tolk, 2012). The SEDEP added “User’s Need Analysis” into the FEDEP. Such an effort is not part of the FEDEP and DSEEP. The features and weaknesses of FEDEP, SEDEP, and DESEP are summarized in Table 18.

In conclusion, the new M&S development process is needed to complement the FEDEP, SEDEP and DESEP, and we can achieve semantic interoperability through it.

Table 18: Comparison of FEDEP, SEDEP and DSEEP

Method	Features	Lacks
FEDEP	• It includes process definition intended for HLA.	• The management aspect of Coordination and control is not addressed sufficiently during the development process of the Federation (Tolk, 2012). • The derived objective was not emphasized from user's requirements. • It focused on federation development in only HLA based environments. • It did not support multiple SSAs.
SEDEP	• It includes process definition for synthetic environments. • The driving objective was emphasized	• It focused on federation development in only HLA based environments. • It did not support multiple SSAs
DSEEP	• It supports including HLA the diversity of SSA such as DIS and TENA. • It supports heterogeneous simulation events.	• The driving objective was not emphasized from user's requirements such as FEDEP. • The assumption of DSEEP is that only one SSA will be used.

4.7.4 Lesson Learned 4: Need for Computer Generated Forces (CGF) (or Semi-Automated Forces (SAFs))

Lesson learned 4, the need for CGFs (or SAFs) was derived from problem 4, limited capability of the CGFs. In this section, we reviewed key factors that determine the performance of the CGFs.

In the last few years, the CGF with Artificial Intelligence (AI) communities has been developing M&S to make synthetic combat environments more realistic. However, in M&S developing communities, current CGF level may seem like a simple automated act in accordance with the prescribed rules and is under the control of the human operator.

Therefore, it is necessary to study the high autonomy of CGF. In other words, the entities are required to have cognitive and automated capabilities to describe human thoughts and the human decisions-making processes by combining both logical and emotional personality characteristics.

4.7.4.1 CGFs Comparison of Simulation Systems

Abdellaoui, Taylor, and Parkinson (2009) conducted a comparative analysis of several existing simulation systems with the CGF tool. Among them, I show the evaluation results of three representative simulation systems as summarized in Table 19. The three products scored higher than other products.

Table 19: CGF Comparison between OneSAF, VR-Forces and STAGE

Category	GOTS	COTS	
	OneSAF	VR-Forces	STAGE
Autonomous Operations	71%	82%	86%
Learning	33%	33%	25%
Organization	55%	55%	52%
Realism	83%	74%	83%
Architecture	71%	71%	63%
Overall Product Score	**69%**	**70%**	**70%**

In Table 19, the evaluation criteria are classified into five categories: autonomous operations, learning, organization, realism, and architecture.

● Autonomy

Autonomy is the ability of a CGF entity to act rationally without human intervention.

● Learning and Adaptation

Learning and Adaptation is the ability of a CGF entity can learn and adapt, to act appropriately by human-directed training.

● Organization

Organization is the ability of a CGF unit-level to perform a team-level activity.

● Realism

Realism is the ability of a CGF entity to act as humans behave.

● Architecture

Architecture covers the arrangement of the CGF entity, external interface and technical support.

VR-Forces scored highest for architecture, and STAGE scored high in realism and autonomy. OneSAF also scored high in realism and architecture. As a result, OneSAF, VR-Forces and STAGE all evaluated well and were satisfactory. Abdellaoui et al. (2009) evaluated VR-Forces as the overall winner.

In conclusion, M&S developing communities need to benchmark the AI techniques of VR-Forces. CGF (or SAF) shall be developed realistically and practically to support analysis, experiment, R&D and training for LVC simulations.

4.7.5 Lesson Learned 5: Need for Multiple SSAs Compliancy

Lesson learned 5 is the need for multiple SSAs compliancy. It was derived from problem 3, the lack of scalability and interoperability of HLA federation. This section covers that simulation systems are necessary to be compliant to multiple SSAs.

HLA is a de facto standard for now because HLA is an IEEE (1516) and NATO standard, and widely used all around the world. Accordingly, most simulation systems are usually compatible with HLA, but some simulation systems do not support all HLA versions. Therefore, all simulation system shall basically support all the different version of HLA and DIS, including HLA 1.3, HLA 1516, HLA 1516e and is compatible with any compliant RTI software such as RTI NG Pro, MÄ K RTI, Pitch pRTI or etc.

Basically, military distributed simulation systems shall have a high degree of interoperability through DIS and HLA for integrating Virtual and Constructive simulation system. Figure 83 and 84 shows the HLA and DIS interface of OneSAF respectively.

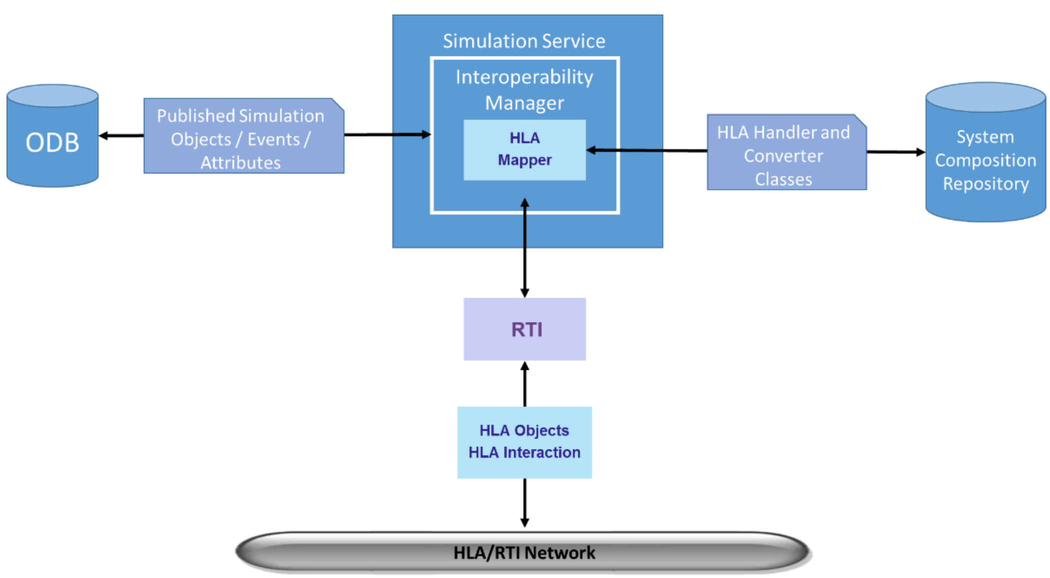


Figure 83: HLA Interface of OneSAF

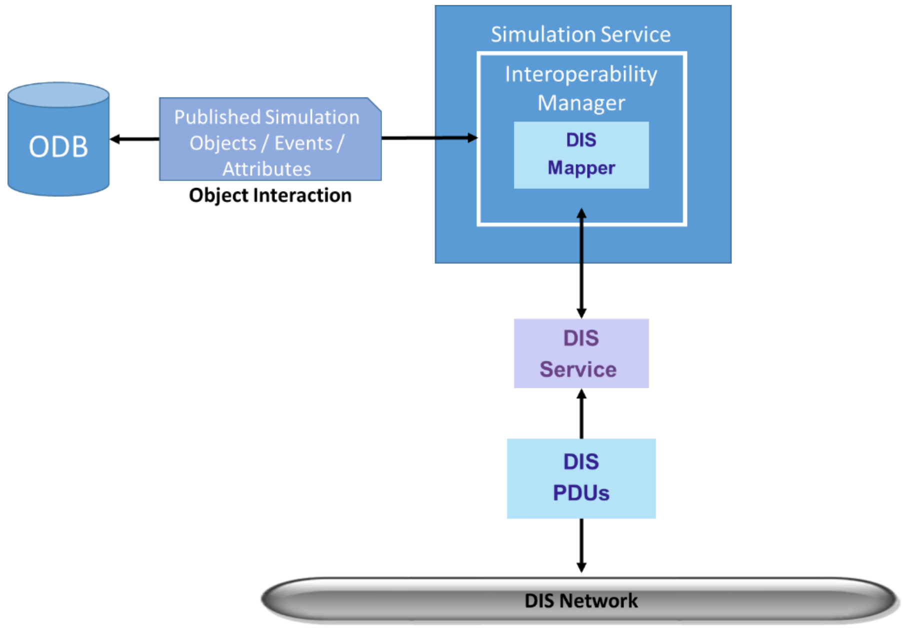


Figure 84: DIS Interface of OneSAF

HLA and DIS compliancy may be sufficient to meet current requirements from M&S developing communities. However, it does not mean that it will always be true. Standards Simulation Architectures (SSAs) are necessary for simulation systems to increase reusability and interoperability. As mentioned before, some SSAs such as DIS, HLA, TENA and CTIA are developed to enable interoperability and reusability of simulation systems. Therefore, in addition to HLA and DIS compliancy, the simulation system shall easily be interoperable to simulation systems based on other SSAs like CTIA, TENA, etc.

In conclusion, the simulation system shall be developed that should be customizable to make it to work with the multiple SSAs other than HLA and DIS.

4.7.6 Lesson Learned 6: Need for Scalability Capability of Simulation Systems

Lesson learned 6 is the need for scalability capability of a simulation system. It was derived from problem 3, the lack of scalability and interoperability of HLA federation.

Simulation systems should be able to achieve a scalable, cross-platform runtime performance simulation in both logical-time and real-time execution modes operating on every mainstream parallel and distributed computing platforms and networks. For this, the simulation systems shall provide workload distribution capabilities to execute large-scale simulation environments because performance of a single computer may not be sufficient to execute the whole LVC simulation. That is, the workloads should suitably be dispersed, the bottlenecks should be removed, and the redundant operations should be avoided.

In order to solve and understand the scalability issue, there are many factors to consider. We must consider how the independent variables affect the dependent variables. Independent variables include the number of entities, the number of nodes in the simulation system, and/or resolution fidelity. Dependent variables include the memory consumption, run time, message bandwidth, and message throughput. A true simulation system for providing a scalable service must take into account all these factors (Jeffrey S. Steinman, 2013).

Dr. Steinman proposed the embedded grid computing for supporting scalable service during simulation execution. The embedded grid computing technology was represented and computationally intensive event was processed well using parallelism, while simultaneously addressing stochastic load balancing issues commonly occurred in large-scale systems (Jeffrey S. Steinman, 2013).

In conclusion, grid computing can be an important technique to eliminate the bottleneck for the scalable capability of simulation systems.

4.7.7 Lesson Learned 7: Need for a Common Correlated Terrain Databases (TDBs)

Lesson learned 7 is the need for a common correlated terrain databases (TDBs). It was derived from problem 6, limited correlated TDB representation. This section emphasizes the need of common correlated TDBs.

Currently, in M&S developing communities, most LVC simulation systems use different numerical systems to calculate simulated actions such as the line of sight and consumption, etc. that involve digitized terrain. This method exacerbates terrain calculation when combined as a simulation environment (or federation), as each of the respective simulation systems has a different numerical system for interacting with the terrain. In the end, results in a mismatch of terrain data and simulation results will not be able to be trusted.

However, correlated dynamic terrain models will remove the need for translating or regenerating the terrain and support efficient terrain calculations. Therefore, the use of correlated terrain is very crucial for the successful interoperation of two or more simulation systems.

A Synthetic Environments (SEs) data is integrated from a number of source data. The SEs represent a geographical region, including terrain, natural, artificial, marine, air and space for M&S. From a military point of view, the operational and battle space environment also includes man-made artifact, and the natural environment includes the land, maritime, air and space domain. Therefore, military simulation environments should encompass the elements above and in additional, atmosphere environments such as weather and wind that change with much smaller time scales, which are very important factors the developers need to capture in the simulation environmental representation.

If the results of the simulation are valid and not overshadowed by the differences recognizing environmental representations, the terrain model should be sufficiently correlated for semantic data interchange and normal execution of a scenario. In addition, the correlated terrain model must be interoperable with current and future force terrain services and address “fair fight” issues. Using a detailed terrain database, the simulation systems will employ highly realistic representations of the physical environment where weapon systems movements and behaviors can be reproduced to enhance training value.

As mentioned in Section 2.7.5.2, the U.S. Army has been developing SEs through the SE-CORE that is the U.S. Army LVC-IA’s Virtual component. The Database Virtual Environment and Development (DVED) of the SE-Core supports the LVC simulation systems by creating TDBs quickly within a few hours or a few days. The U.S. Army aims to create global SEs within 96 hours.

Therefore, it is important for M&S developing countries to develop a correlated TDB system like the DVED of the U.S. Army.

4.7.7.1 Basic Types of Geospatial information systems (GIS) Data

SE refers to a representation of the spatial dimension that may or may not represent the actual position of the world. The terrain is defined by many possible characteristics that may have to be taken into account when modeling the natural environment (Tolk, 2012). GIS data are the main starting point and are the general formats for building the SE in many simulation systems used for training.

The outcomes of the environmental data collection become source data for both 2D and 3D GIS applications. Then, GIS applications will be configured to SE representation. All multiple GIS data must be made into a usable format that the simulation systems can be perceptible during a simulation execution. For this process, the popular GIS file formats are grid formats, vector formats, raster formats, (for elevation) and other formats (Tolk, 2012). The following subsections describe these formats.

4.7.7.1.1 Raster Data Formats

Raster data formats, are data that are decomposed uniform cells with each cell storing a single value that describes something about the area in the real world. Table 20 summarizes the raster formats.

Table 20: Raster formats

Formats	Descriptions
ADRG	• ARC Digitized Raster Graphics
CADRG	• Compressed ADRG
CIB	• Controlled Image Base
DRG	• Digital raster graphic
ECRG	• Enhanced Compressed ARC Raster Graphics
ECW	• Enhanced Compressed Wavelet
Esri grid	• Environmental Systems Research Institute grid
GeoTIFF	• Tagged Image File Format
IMG	• ERDAS IMAGINE
JPEG	• Joint Photographic Experts Group
MrSID	• Multi-Resolution Seamless Image Database
netCDF	• Network Common Data Form
RPF	• Raster Product Format

4.7.7.1.2 Vector Data Formats

Vector (directional lines) is used to represent a geographic feature. Vector data is characterized by the use of sequential polygons, lines, or points. Each point is represented by the X, Y coordinates. Table 21 summarizes the vector data formats.

Table 21: Vector Data Formats

Formats	Descriptions
AutoCAD DXF	• CAD data file format developed by Autodesk
DLG	• Digital Line Graph
GML	• Geography Markup Language
GeoJSON	• An open standard format for encoding various geographic data structures
GeoMedia	• a geographic information system (GIS) application by Intergraph
KML	• Keyhole Markup Language
MapInfo TAB format	• A geospatial vector data format for GIS software by MapInfo Corporation
NTF	• National Transfer Format
Spatialite	• A spatial extension to SQLite
Shapefile	• A geospatial vector data format for GIS software
Simple Features	• International Organization for Standardization (ISO) 19125 standard
SOSI	• Systematic Organization of Spatial Information
SDF	• Spatial Data File
TIGER	• Topologically Integrated Geographic Encoding and Referencing
VPF	• Vector Product Format is military standard structure by the U.S. Defense Mapping Agency (DMA)

4.7.7.1.3 Grid Data Formats

Table 22 summarizes the grid data formats.

Table 22: Grid Data Formats

Formats	Descriptions
DEM	• Digital Elevation Model
GTOPO30	• A digital elevation model for the world, developed by USGS
DTED	• Digital Terrain Elevation Data • The most popular data are often used in military simulation systems.
GeoTIFF	• Tagged Image File Format
SDTS	• Spatial Data Transfer Standard

4.7.7.2 Terrain Data Formats

We have researched terrain data formats that several simulation systems use. As shown in Figure 85, each of simulation systems typically require specialized formats optimized for simulation execution and visualization, such as a tree, building, and top, etc. Figure 85 shows a sample desert village that illustrates several formats, including Steel Beasts Pro, OpenFlight, VBS2, JSAF, JCATS, MÄ K VR-Forces, OneSAF, OneSAF Testbed, and OpenSceneGraph (OSG).

Figure 85: Terrain formats

Source: http://www2.calytrix.com/support/terrain/overview/

In conclusion, M&S developing communities should develop the common correlated TDBs tool system to support several terrain data formats.

4.7.8 Lesson Learned 8: Need for a New Common Standard Simulation Architecture (C-SSA)

Lesson learned 8 is the need for a new common standard simulation architecture. It was derived from problem 3, the lack of scalability and interoperability of HLA federation.

For the standard simulation architectures (SSAs), all simulation systems use are different. One benefit of having common standard simulation architecture (C-SSA) is that services and models make use of the same programming constructs, and therefore, can be more freely interoperable.

In Section 2.1.1, according to the DoD Modeling and Simulation (M&S) Master Plan, the first objective was to create a common technical framework (or SSA) for M&S development. The HLA fulfilled one of the objectives of the M&S Master Plan partially, but it is not the perfect C-SSA. Therefore, as mentioned earlier, the research about new C-SSA shall be studied continuously.

However, we estimated that development of new C-SSA or convergence of current multiple SSAs is difficult to be realized in the near future.

4.7.9 Lesson Learned 9: Need for a Product Line Architecture Framework (PLAF) Concept

Lesson learned 9 is the need for a Product Line Architecture Framework (PLAF) Concept. It was derived from problem 2, the lack of reusability, and problem 3, the lack of scalability and interoperability of HLA federation.

PEO STRI has accepted the Product Line approach and has been utilizing it to develop new interoperable simulation systems and services. The Product Line approach enables the needs of the Warfighter to respond quickly because they reuse pre-built components and products.

There are product line initiatives within each of the Live, Virtual, and Constructive Domains of PEO STRI. Each domain includes the Live Training Transformation (LT2), Synthetic Environment Core (SE CORE), and the Joint Land Component Constructive Training Capability (JLCCTC) respectively (Faulk et al.).

Prior to the LT2, SE Core and JLCCTC product line in the U.S. Army, to date, most respective Live, Virtual or Constructive training simulation systems have been developed separately by a variety of different manufacturers in the U.S. Each simulation system has been developed with a single purpose, its own architecture framework, software and components in U.S.

Even now, the developing M&S country (or community) such as South Korea has been developing the simulation systems using a conventional system development approach without a common and standard concept.

Therefore, newly developing M&S countries (or communities) shall make a common simulation architecture framework such as the PLAF to avoid repeating the same mistake the U.S. Army has experienced. Through successful development of the common simulation architecture framework strategy, each of Live, Virtual and Constructive common simulation architecture will provide a set of common components that support integrated and interoperable training solutions for LVC simulation.

4.7.10 Lesson Learned 10: Need for a Simulation System to Support Multiple M&S Applications and LVC simulations

Lesson learned 10 is the need for a simulation system to support multiple M&S applications and LVC simulations. It was derived from problem 7, the limited use of the simulation systems. Simulation systems can be classified such as Training, Analysis, R&D and Acquisition based on the objective. However, the development of the simulation system just to meet a user requirement’s one objective is a backwards move that contradicts the interoperability and reuse of M&S.

Therefore, simulation system must be able to support multiple M&S applications such as the research and development (R&D); development and acquisition; decision making support; engineering experiments; testing and evaluating (TE), analysis; and training, exercises and military operations in order to promote interoperability and reuse. In addition, simulation systems should be able to support a Live, Virtual and Constructive simulation systems. In other words, the simulation system shall be designed with flexibly to serve multiple objectives.

Among the existing legacy simulation systems, One Semi-Automated Forces (OneSAF) has extendable capabilities to provide comprehensive support of emerging the U.S. Army functional requirements and technical standards and is being developed as a standard for Constructive simulation. OneSAF is the U.S. Army’s next-generation simulation system being developed based on the PLAF concept to provide an integral simulation service to the Advanced Concepts and Requirements (ACR), Training, Exercises, and Military Operations (TEMO), and Research, Development, and Acquisition (RDA) domains (Wittman Jr & Harrison, 2001).

In conclusion, in order to develop the simulation system that supports multiple M&S application as well as Live, Virtual and Constructive simulations, it is necessary to develop a *Common Standard Simulation Architecture Framework.*

4.7.11 Lesson Learned 11: Need for a Battle Damage Assessment (BDA) Application

Lesson learned 11 is the need for a battle damage assessment (BDA) application. It was derived from problem 8, limited analysis of engagement result.

Usually, when we evaluate or analyze the effectiveness of weapon systems, tactics or operations for a combat situation, we refer to the Measure of Performance (MOP) and Measures of Effectiveness (MOE). MOP measures the performance of specific parameters in terms of engineering level. MOE is the measure of the degree to accomplish the mission under the conditions given weapon system in terms of engagement level. For analysis, effectiveness or evaluation, the weapon systems or echelon battalion and below is often modeled at the engagement with a standard combat scenario and parametric weapon information.

4.7.11.1 Aircraft Combat Survivability (ACS)

In the case study, we demonstrated the Air-Defense Engagement scenario, but we could not analyze the BDA in detail due to the absence of application for the BDA. Therefore, in this section, we provide the mathematical concepts for Aircraft Combat Survivability (ACS) analysis.

ACS is defined as the capability of an aircraft to avoid or withstand a man-made hostile environment. As a consequence of the uncertain nature of an unpredictable combat, aircraft survivability is measured by probability. The probability that aircraft will survive is denoted as P_S.

The probability the aircraft will be killed or destroyed is denoted as P_k. Therefore, the probability P_s is the complement of P_k (Ball, 2003). Thus, the formula is as follow:

$$P_s = 1 - P_k \tag{1}$$

4.7.11.1.1 One on One Scenario

The one-on-one scenario can be divided in two parts: One is the susceptibility part and the other one is the vulnerability part. The susceptibility part can be divided into five sequential phases. Within each phase there are one or more operational functions that must be performed by the various elements of the air defense. In order to hit the aircraft, the threat weapon such as SAM should do the following:

First, the threat weapon searches for the aircraft. Second, it detects the aircraft using a radar. Third, it engages the aircraft by firing a gun or launching a missile to the aircraft. Fourth, the gun-fired ballistic projectile or the guided missile from the threat weapon, both known as the threat propagator, must "fly out" and intercept the aircraft. Fifth, the damage mechanisms carried by the warhead on the propagator must hit the intercepted aircraft, either by a direct hit or by a proximity fuzing. Finally, the damage mechanisms that hit the aircraft must kill one or more of the aircraft's critical components, resulting in the loss of an essential function for flight or mission completion (Ball, 2003). Figure 86 illustrates the tree diagram for the one-on-one Scenario (Single Shot).

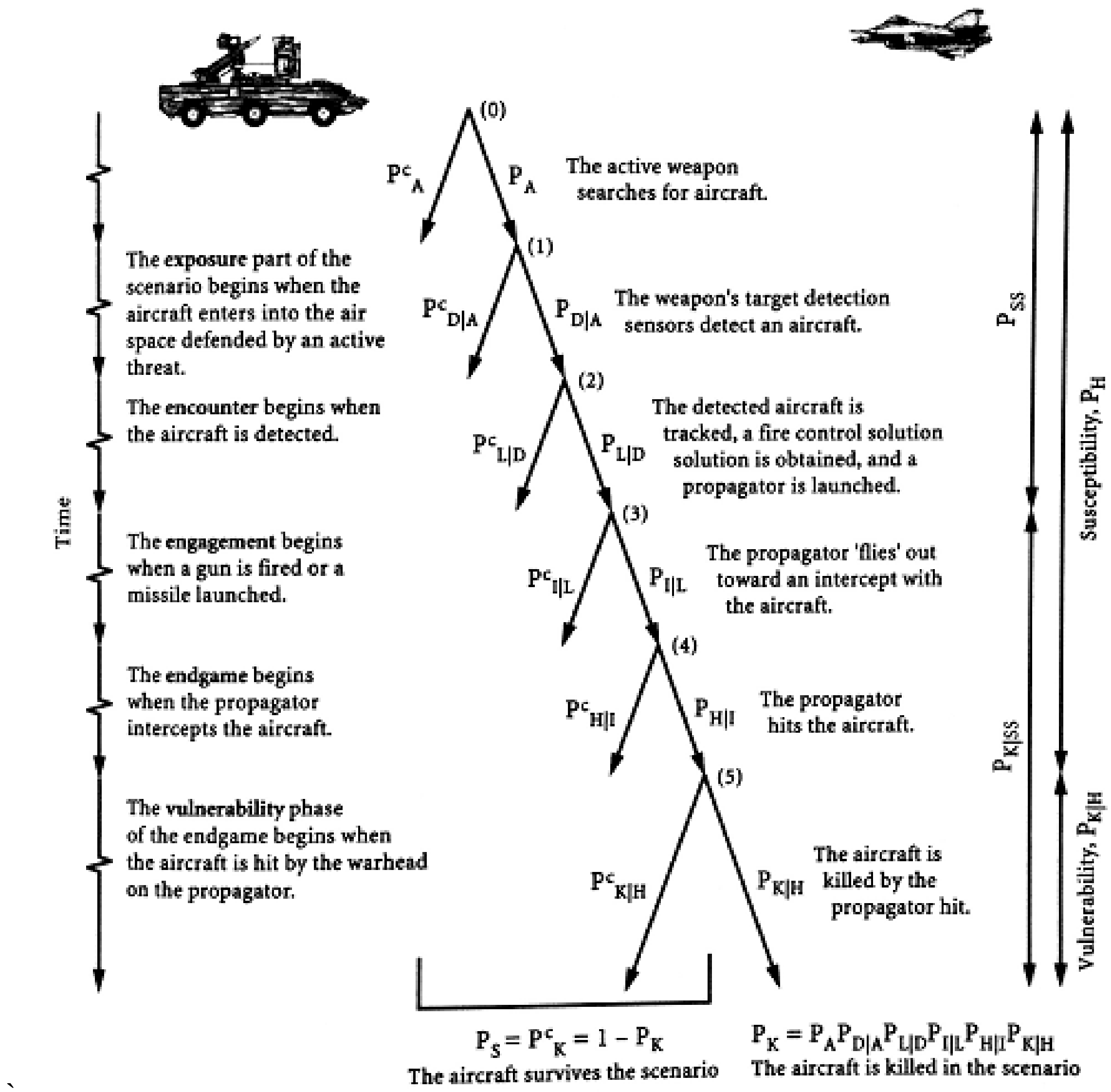


Figure 86: Tree Diagram for the One-On-One Scenario (Single Shot)

Source: Ball (2003)

4.7.11.1.2 Probabilities

This section explains the probabilities describing the figure above (Ball, 2003).

● P_A is the probability that the threat weapon is active as the aircraft approaches the threat weapons, in other words, the weapon is actively or passively searching, and ready to encounter and engage flying aircraft within its defense area.

● $P_{D|A}$ is the conditional probability that the aircraft is detected, given that the threat weapon is active.

● $P_{L|D}$ is the conditional probability that the aircraft is tracked, a fire control solution is obtained, and a missile is launched or a gun is fired to the aircraft, given that the threat weapon was active and detected the aircraft.

● $P_{I|L}$ is the conditional probability that the threat propagator approaches or intercepts the aircraft, given that the propagator was launched or fired to the aircraft.

● $P_{H|I}$ is the conditional probability that the propagator hits the aircraft, given that the propagator has intercepted the aircraft.

● $P_{K|H}$ is the conditional probability that the aircraft is killed or destroyed, given a direct hit by the propagator.

4.7.12 Lesson Learned 12: Need for a General Bridging Tool

Lesson learned 12 is the need for a general bridging tool. It was derived from problem 3, the lack of scalability and interoperability of HLA federation. This section covers the latest technologies for Interoperability.

In the case study, bridging tool was needed because it is not practical to get every asset to agree on a protocol, HLA 1.3 FOM RTI, HLA 1516 FOM RTI, HLA 1516e FOM RTI, DIS 2.0.4, IEEE 1278.1, IEEE 1278.1a PDUs, or TENA LROM. In Section 2.5, we reviewed several bridging solutions such as gateway, middleware, broker, and protocol solution.

In general, bridging tools are demanded whenever it is impossible to achieve direct interoperability among a set of different simulation systems which are not compliant to multiple SSAs such as, DIS, HLA, TENA and CTIA. In other cases, bridging is needed because a system architect wants to implement a hierarchical federation of federations design. Bridging is often needed to support large-scale LVC simulation environment, or to support interoperability of a simulation system to C4I system (MÄ K).

A desirable bridging should provide a simple bridging function between two simulation systems, or can be used to support a more complex federation of federations architecture, where multiple, heterogeneous assets are interconnected to support large-scale LVC simulation (MÄ K).

We have identified the latest bridging solution technology from M&S market to support a more effective LVC interoperation using bridging tool. The identified the software tool is VR-Exchange by VT-MÄ K. This software tool will assist users and developers in the discovery and development of a common bridging solution for future LVC simulation environments.

4.7.12.1 VR-Exchange

VR-Exchange (or universal translator) is a bridging software tool developed by the VT-MÄ K for heterogeneous distributed simulation environments. VR-Exchange allows simulations that use incompatible SSAs to interoperate, regardless of whether they use the same FOMs and RTIs. For example, within the HLA world, using VR-Exchange, federations using the HLA RPR FOM 1.0 can interoperate with simulations using RPR FOM 2.0, or federations using different manufactured RTIs can interoperate (MÄ K).

In addition, VR-Exchange supports HLA, TENA, and DIS translation and enables heterogeneous simulation environments to interoperate. VR-Exchange brokers accommodate different FOMs and LROMs using FOM Mappers and LROM Mappers. Mappers are dynamic link libraries that map the objects and concepts of a particular FOM or LROM.

VR-Exchange consists of a portal and brokers. VR-Exchange permits simulations to interoperate through the use of a shared memory space (Portal) and brokers (See Figure 87). The Portal of VR-Exchange is a web site that provides access to all federates contained in the federation. Each broker of VR-Exchange translates between its native SSA protocol and the VR-Exchange common simulation representation. The translated data passes through the Portal and is translated by the other brokers to their protocol. Figure 87 shows three different federations. Each uses the broker, but each broker is configured to use different RTIs and different FOMs. Each unique broker configuration is called a connection.

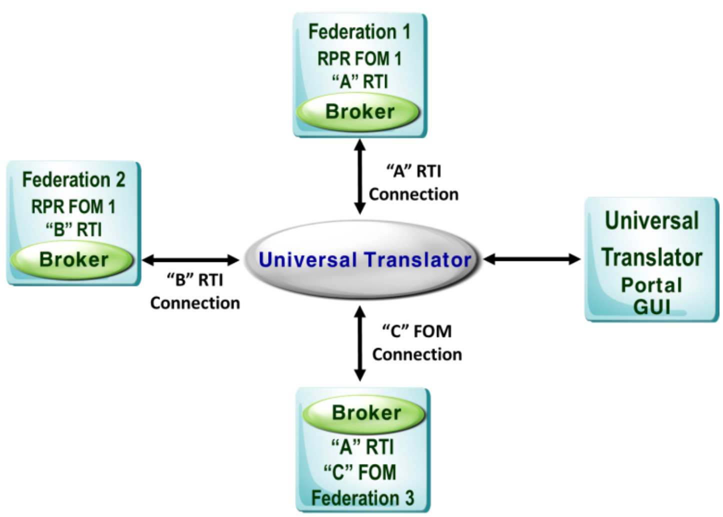


Figure 87: VR-Exchange (Universal translator) Architecture

4.7.12.2 Broker

This section describes the main features of the broker. The broker is a software application that is a translator for distributed simulations. Each broker has translators for the object or interaction classes that it supports. The demanding technology for bridging between federations is a broker. The broker allows a user to combine a number of sub federation into a large federation.

VR-Exchange includes brokers for HLA, TENA, and DIS. It has HLA brokers for the HLA 1.3 specification, for IEEE 1516 specification, and for the HLA Evolved (IEEE 1516-2010) specification. The HLA brokers support the versions of the RPR FOM. The TENA broker allows VR-Exchange to participate in TENA executions. It uses LROM Mappers to map TENA objects to the VR-Exchange common simulation representation. The DIS broker receives DIS 2.0.4, IEEE 1278.1, and IEEE 1278.1a PDUs.

In conclusion, M&S developing communities need to develop these technologies over the benchmark, and these technologies will improve the interoperability between existing simulation systems.

4.8 Phase 6: Recommended Actions

This section summarizes the major recommended actions. To realize these lessons learned we have developed the following set of recommended actions. These actions are listed in priority order. The lessons learned from the case study helped to construct a set of recommended action that should be applied to the agile roadmap of the LVC-ITA. Table 23 is an overview list of the recommended action that must be one step among the roadmap for LVC-ITA.

Table 23: List of the Recommended Actions

From Lessons Learned No.	Recommended Actions No.	Description
Lesson Learned 1 Lesson Learned 2 Lesson Learned 3 Lesson Learned 4	Recommended Action 1	• Common Standard- Defense Modeling and Simulation Process (CS-DMSP)
Lesson Learned 9 Lesson Learned 10 Lesson Learned 11	Recommended Action 2	• Common Standard – Simulation System Architecture Framework (CS-SSAF)
Lesson Learned 7	Recommended Action 3	• Common Standard-Correlated Terrain Database (CS-CTDB)
Lesson Learned 5 Lesson Learned 6 Lesson Learned 8 Lesson Learned 12	Recommended Action 4	• Advanced Interoperability Technology

Figure 88 shows the overall flow to the recommended action from the finding problems in the case study.

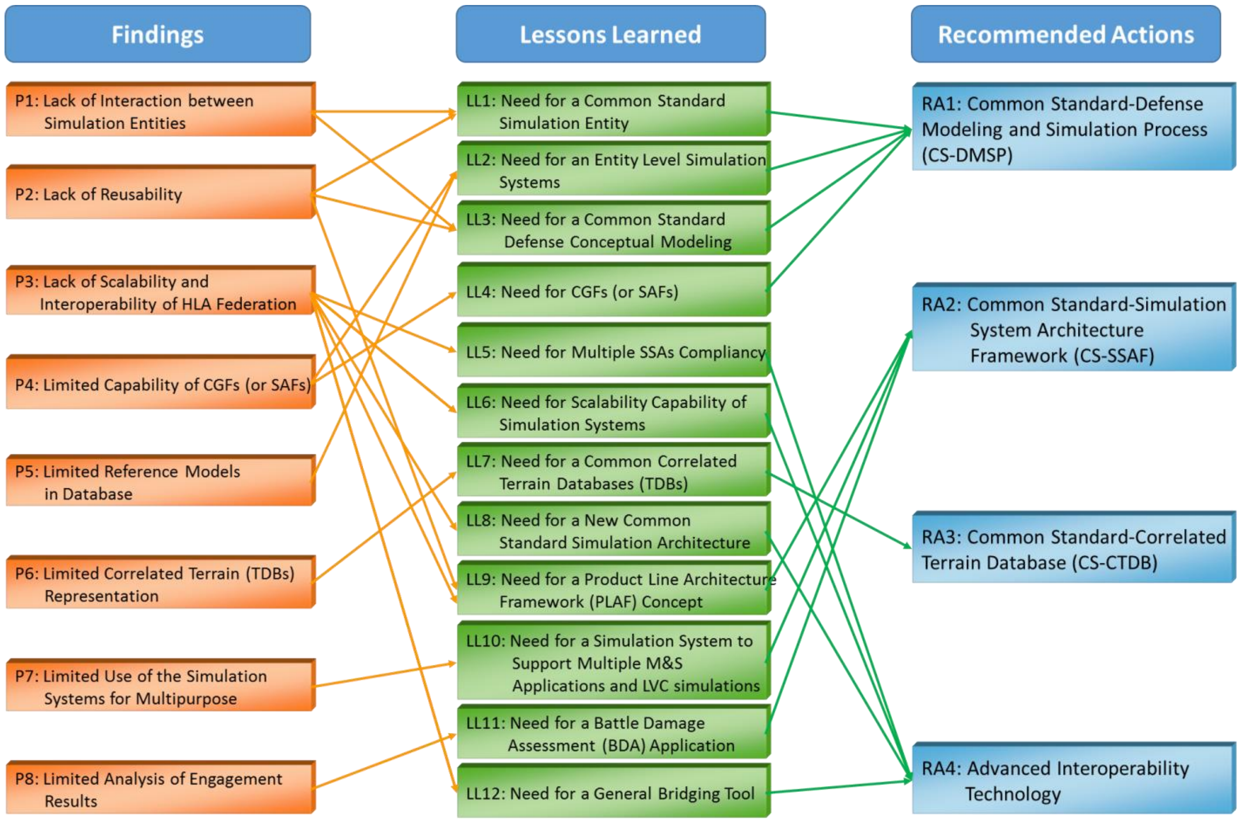


Figure 88: Recommended Actions from Case Study Findings and Lessons Learned

CHAPTER FIVE: AGILE ROADMAP FOR LVC-ITA

This chapter covers the agile roadmap for LVC-ITA. We found what was the lacking from the case study results and discussed these limitations in the previous section. These limitations are connected to the LVC-ITA issue and must be addressed, then solved. Afterwards, we drew lessons learned. These lessons learned are connected to recommended actions (or requirement) for the LVC-ITA roadmap.

Through the LVC simulation case study and literature review, the agile roadmap consists of four recommended actions. They are (a) Common Standard-Defense Modeling and Simulation Process (CS-DMSP), (b) Common Standard-Simulation System Architecture Framework (CS-SSAF), (c) Common Standard-Correlated Terrain Database (CS-CTDB), and (d) Advanced Interoperability Technology.

The agile roadmap for LVC-ITA is needed for the M&S developing country (or community) to avoid the process of trial-and-error U.S. DoD has undergone and to develop M&S environment systematically. The agile roadmap for LVC-ITA covers multiple related topics, but the roadmap did not examine any particular topic thoroughly. The agile roadmap for LVC-ITA tries to provide technically feasible, affordable, implementable, and right solutions and guidelines associated with each topic.

M&S community can achieve reuse, interoperability, and composability when we complete each recommended action step by step. First, we expect to achieve the establishment of M&S development process and the enhancement of entity (or object) model interoperability when M&S community completes the CS-DMSP. Second, we expect to achieve the enhancement of reuse and composability when M&S community completes the CS-SSAF. Third, we expect to achieve supporting correlated terrain database for LVC simulation system when M&S community completes the CS-CTDB. Lastly, we expect to achieve supporting interoperability between LVC simulation systems when we complete the advanced interoperability technology. The agile roadmap for LVC-ITA and its expectations are summarized in Table 24.

Table 24: Overview of Agile Roadmap for LVC-ITA

RA No.	Recommended Action (RA)	Expectation
RA1	Common Standard-Defense Modeling and Simulation Process (CS-DMSP)	• Establishment of M&S Development Process • Enhancement of Entity (or Object) Model Interoperability
RA2	Common Standard-Simulation System Architecture Framework (CS-SSAF)	• Enhancement of Reuse and Composability
RA3	Common Standard-Correlated Terrain Database (CS-CTDB)	• Supporting Correlated Terrain Database for LVC simulation system
RA4	Advanced Interoperability Technology	• Supporting Interoperability between LVC simulation systems

5.1 Recommended Action 1: Common Standard - Defense Modeling and Simulation Process (CS-DMSP)

Recommended Action1, CS-DMSP is recommended to realize (1) lesson learned 1, the need for a common standard simulation entity, (2) lesson learned 2, the need for an entity level simulation system, (3) lesson learned 3, the need for a common standard-defense modeling, and (4) lesson learned 4, the need for CGFs (or SAFs). This section covers the CS-DMSP and common model. The following subsection describes the CS-DMSP and a common model.

5.1.1 Common Standard-Defense Modeling and Simulation Process (CS-DMSP)

This section covers the Common Standard-Defense Modeling and Simulation Process (CS-DMSP). We reviewed several existing M&S approaches related with the conceptual modeling (CM) in Section 2.4. The approaches are Federation Development and Execution Process (FEDEP), Synthetic Environment Development and Exploitation Process (SEDEP), Distributed Simulation Engineering and Execution Process (DSEEP), Conceptual Models of the Mission Space (CMMS), and Defense Conceptual Modeling Framework (DCMF).

The U.S. DoD Modeling and Simulation (M&S) Master Plan established CMMS as the second component of the M&S Common Technical Framework. Because the CMMS is the common starting point and eventual real-world baseline for consistent and authoritative M&S representations, conceptual modeling is undoubtedly the most important aspect of military M&S development. Military M&S community often requires large-scale LVC simulation environments. Therefore, there is much interest in model reuse and distributed simulation.

Because of the importance of the conceptual modeling, I developed the CS-DMSP on a basis of DSEEP, SEDEP, and DCMF. The DSEEP developed from FEDEP and SEDEP is recommended as practice documents describing how to develop and implement a simulation environment. The SEDEP improved that the FEDEP was usually driven by a "technical" need and viewpoints (Tolk, 2012). As a result, the SEDEP added "user's need analysis". Such an effort is not part of the FEDEP and DSEEP. The DCMF improved on the conceptual analysis of the CMMS.

Therefore, we offer the mixed process for the defense conceptual modeling using strength of DSEEP, SEDEP and DCMF respectively as shown in Figure 89. I named the mixed process a Common Standard-Defense Modeling and Simulation Process (CS-DMSP). The approach discussed here is provided for developing the conceptual model of the mission space for M&S developing community in military M&S area. The CS-DMSP consists of nine steps on the top level.

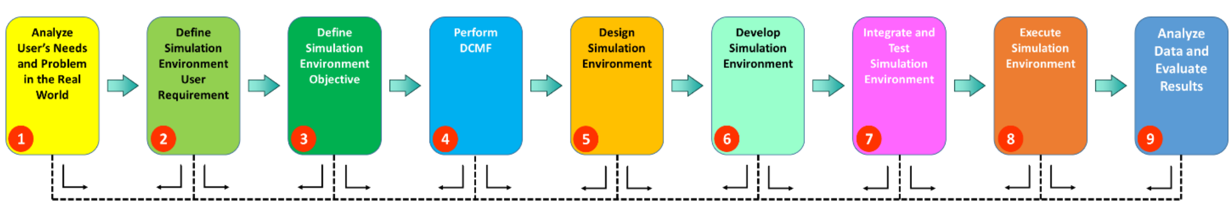


Figure 89: Common Standard-Defense Modeling and Simulation Process (CS-DMSP)

Each step of the CS-DMSP is described in detail.

- Step 1: Analyze User's Needs and Problems in the Real World (Ford, 2005)

Step 1 is developed from Step 1 of SEDEP and an additional step as the start of the process, in comparison with FEDEP and DSEEP. The purpose of Step 1in the CS-DMSP is to understand user's needs and problems without any influence from the environment from a high level view (Not technical). In other words, problem recognition is not affected from the simulation environment. The effort of this step is not found in FEDEP nor DSEEP.

● Step 2: Define Simulation Environment User Requirement (Ford, 2005)

Step 2 is also developed from Step 2 of SEDEP. The purpose of Step 2 in the CS-DMSP is to provide a comprehensive description of what the problem setter(s) wants from the simulation environment. This is achieved by the problem setter and problem solver working together to define the simulation environment user requirements. In this step, evaluating the objectives and defining the scenario are performed in terms of operational view (Not technical) (Ford, 2005). The FEDEP and DSEEP start after this step.

● Step 3: Define Simulation Environment Objective (IEEE, 2003, 2011).

Step 3 is developed from Step 1 of FEDEP and DSEEP. The purpose of Step 3 in the CS-DMSP 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 from a technical view.

● Step 4: Perform DCMF (Mojtahed et al., 2005)

In order to reinforce Step 2 of the FEDEP and DSEEP which is "Perform conceptual analysis" and the Step 2 of the SEDEP which is "Define Federation System Requirements," ,which the DCMF replaced. The purpose of Step 4 in the CS-DMSP is to develop an appropriate representation of the real military domain that applies to the defined problem space and to develop the appropriate military operation scenario from a system and technical view.

● Step 5: Design Simulation Environment (Ford, 2005; IEEE, 2011)

Step 5 is developed from Step 3 of the DSEEP and Step 3 of the SEDEP. The purpose of Step 5 in the CS-DMSP is to create the design of the simulation environment that will be implemented in Step 8. The technical specifications for simulation environments are agreed upon from a system and technical point of view.

● Step 6: Develop Simulation Environment (IEEE, 2011)

Step 6 is developed from Step 4 of the DSEEP. The simulation data exchange model (SDEM) is developed, simulation environment agreements are established, and new simulation systems (e.g. simulations, simulators, databases, data loggers, network infrastructure etc.) either with or without modifications to existing simulation systems are implemented from a system and technical point of view.

● Step 7: Integrate and Test Simulation Environment (Ford, 2005; IEEE, 2011)

Step 7 is developed from Step 5 of the DSEEP and Step 5 of the SEDEP. The purpose of Step 7 in the CS-DMSP is to configure and integrate the simulation environment. Integration activities are performed, and testing is conducted to verify that interoperability requirements are being met.

● Step 8: Execute Simulation Environment (Ford, 2005; IEEE, 2011)

Step 8 is developed from Step 6 of the DSEEP and Step 6 of the SEDEP. The purpose of Step 8 in the CS-DMSP is to prepare the simulation environment for execution, to run the simulation environment scenario, and to collect and preprocess the output data from the execution for performing the evaluation.

● Step 9: Analyze Data and Evaluate Results (Ford, 2005; IEEE, 2011)

Step 9 is developed from Step 7 of the DSEEP and Step 7 of the SEDEP. The purpose of Step 9 in CS-DMSP is to analyze the output data acquired from the simulation environment execution and evaluate the results, which are reported back to the problem setter, user or sponsor to decide if the problem being investigated has been solved or further work is required.

5.1.2 Common Model

This section describes a common model. In establishing an LVC-ITA, a major challenge is to determine how run-time simulation data is to be aligned and shared across the heterogeneous LVC domains, as well as how simulation objects in the LVC domains will interact, both syntactically and semantically.

In the simulation world, an entity is a single object of any type. Each entity acts as a channel for information, holding pointers to callback functions that retrieve information regarding the entity. Each entity contains a list of attributes and a list of actions such as the ability to move, shoot, communicate and more.

All military operations or work might be special and unique, but a number of processes can also be supported by common standard solutions. Therefore, we need to find a common similarity between many entities and between numerous military operations. Thus, if we use a common entity and common operational model which the M&S community developed, developers can easily develop and modify the model using a common combat operation and common combat objects when modeling the new entity and the new military operation. Therefore, we need to build a repository of generalized concepts on military operations and combat objects.

Today, armed forces must operate in coalition forces, task forces, and joint operations where unit and equipment performance varies widely. All of these different force and equipment mixtures create the need for simulation systems that can handle multiple forces with varying equipment and capabilities.

There is a common object that can be used and required in the other simulation systems. Simulation systems shall provide a model library which contains readily available simulation entities. Each of the simulation entities shall be easily utilized in different simulation systems depending on the nature of the system. The simulation system shall also enable the addition of new models to the library. The model library shall provide predefined models such as: platform models (air, ground, sea, etc.), air model, ground model, and sea model (Ç elik et al., 2012), or friendly, opposing, neutral, and so on.

5.1.2.1 Conceptual Model of Common Combat Entity

The players in a simulation are called entities. There are two broad classes of entities: (a) single entity and (b) unit. Singular entity level models model the physical phenomenology of interest in the level of individual entity (Tolk, 2012).

For modeling common combat entity, we need to identify what generalizations are needed to describe combat entities. Combat entities can act in and respond to their environment. The main question is: "What characteristics should be included in modeling combat entities"? We identified what the most basic characteristics are from VR-Forces, SIMbox and AddSIM in

SIL. These are abilities to move, shoot, look, communicate and weapon. Thus, a generalization of the essential characteristics of combat entities can provide an easy way to describe many different combat units.

For example, there are also different types of combat units that need to be modeled. An infantry platoon is obviously different from a tank platoon, but they do have similar characteristics. For example, moving, shooting, and communicating are features that occur in every one of them as shown in Figure 90.

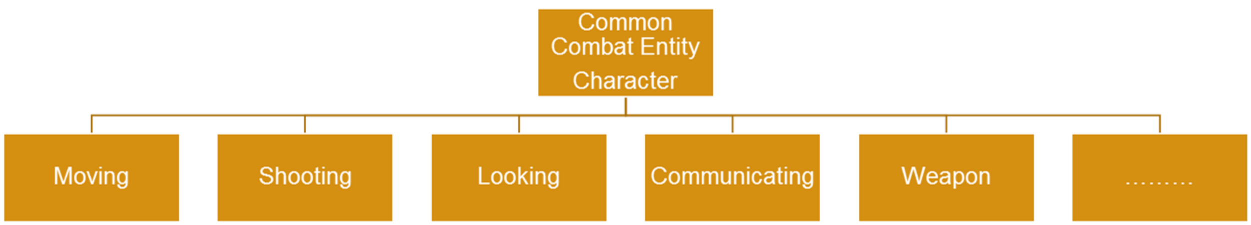


Figure 90: Common Characters of Common Combat Entity

5.1.2.2 Conceptual Model of Common Combat Operation

Each entity is part of a force-level organizational unit. Since the unit consists of entities, they have most of the characteristics of single entities as described in the previous section.

For modeling common combat operation, we need to identify what characteristics are needed to describe the many different types of military ground combat operations as shown in Figure 91.

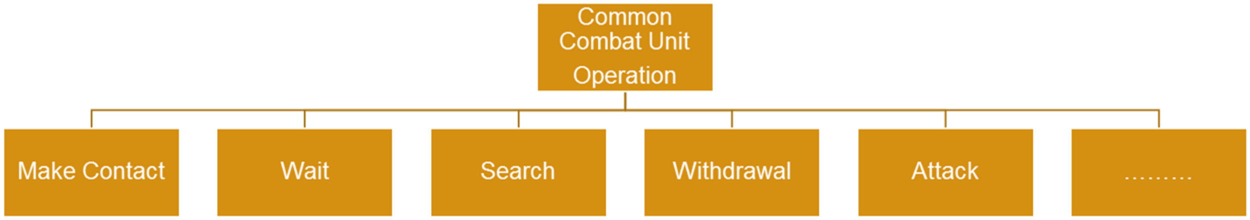


Figure 91: Common Operations of Common Combat Unit

5.1.2.3 Computer Generated Forces (CGF)

CGF should be developed in the following way. In particular, CGF such as a soldier and small unit of a human model, must be represent more realistic in representing of entity behaviors and unit operation. CGF can be developed based on a common model when modeling good rational or cognitive models within a CGF in a military operational environment.

When CGF receives the impact from an environmental factor, including the complexity of the combat operational environments and the weather or when CGF's internal states are changed due to the physiological factors such as workload and psychological stress factors, the CGF will behave like a common combat entity character. Small unit will also behave like a common combat unit operation. That is, the reactions are automated behaviors which are run as a result of situational conditions within the CGF. With more detailed common modeling and composite common modeling, characteristics of human behavior can be modeled.

5.2 Recommended Action 2: Common Standard-Simulation System Architecture Framework (CS-SSAF)

Action 2 is recommended to realize: (a) lesson learned 9, need for a Product Line Architecture Framework (PLAF) concept, (b) lesson learned 10, need for a simulation system to support multiple M&S applications and LVC simulations, and (c) lesson learned 11, need for a battle damage assessment (BDA) application. This section covers the CS-SSAF.

We have identified the methodologies and technologies needed for a seamless LVC simulation. If all of these technologies are included as a component in the LVC simulation systems architecture framework, a seamless LVC simulation will be realized.

As mentioned in Section 2.7.5, the U.S. Army LVC-IA has three major components LT2-FTS, SE Core and JLCCTC. Each of these components owned an architecture framework that can be referenced when developing Live, Virtual and Constructive simulation systems. LT2-FTS has a common plug-and-train components called the LT2 Component PLAF. SE-Core has the Virtual Simulation Architecture (VSA) PLAF. Similarly, JLCCTC has the JLCCTC Objective Architecture and OneSAF architecture.

The PLAF is intended to identify the basic components, products, and interfaces that support the entire simulation system requirements. It also relates a set of guiding principles for the product line based architecture. It is envisioned that the simulation system developer can revise and extend the PLAF to become the formal Product Line Architecture Specification (PLAS) that fully specifies the architectural components, products, interfaces, and services (Wittman Jr & Harrison, 2001).

Therefore, we propose that each of the common standards Live, Virtual and Constructive simulation systems architecture frameworks can be used in the Army, Navy and Air-Force. The Common Standard- Simulation Architecture Framework (CS-SSAF) is intended to identify the basic components, products, and interfaces that support the entirety of Live, Virtual and Constructive simulation systems.

The CS-SSAF is a set of tools, data, and components for assembling simulation system for training, analysis and acquisition interoperable with Live, Virtual and Constructive simulation systems. The CS-SSAF will contribute to increase interoperability between training simulation systems, to increase the reuse of products developed for training systems, to save on the developing cost and total life-cycle cost. In addition, the CS-SSAF will be a main part that supports the LVC-ITA.

5.2.1 Common Standard - Simulation System Architecture Framework (CS-SSAF)

This section describes the overall Common Standard-Simulation System Architecture Framework (CS-SSAF). We have researched the requirements for Live, Virtual and Constructive simulation systems from literature review, the simulation systems we hold and M&S communities. After harvesting all the requirements, we have developed the CS-SSAF baseline. Figure 92 shows the CS-SSAF.

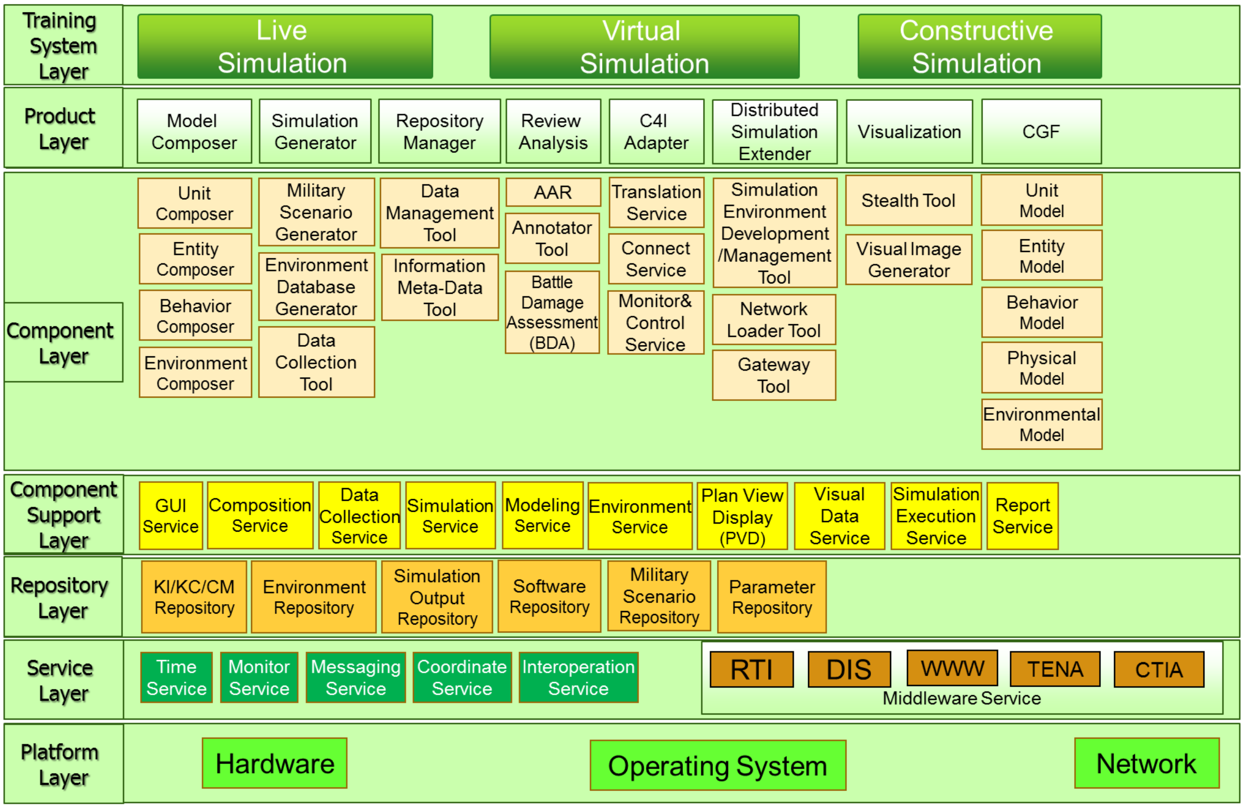


Figure 92: CS-SSAF

CS-SSAF was designed in the layered architecture for prevention against duplicated functions at each layer, ease of maintenance and convenience in developing models. The architecture consists of a training system layer, product layer, component layer, component support layer, repository layer, service layer including middleware service, and platform layer.

In the following subsections, I describe the products and components of each layer by referring to OneSAF: a product line approach to simulation development in CS-SSAF (Wittman Jr & Harrison, 2001).

5.2.1.1 Training System Layer

The training system supports Live, Virtual and Constructive simulation system. It constitutes a training system assembly that meets specific training and combat experimental requirements.

5.2.1.2 Product Layer and Component Layer

The product layer is given to show the set of multiple products necessary to form a complete system configuration. The products are stand-alone. Each product is a composed of several components that need to be developed or harvested through reuse to support the product. Components are systematically reusable building blocks of products and can be an independent executable tool or model (Wittman Jr & Harrison, 2001).

5.2.1.2.1 Model Composer Product

The Model Composer Product supports the creation of entities, actions, or environmental factors from a collection of primitive components. Metadata associated with each primitive component constrains the process in the creation of allowable constructs. At a system level, the composer supports the creation of tailored applications from desired software modules or artifacts. Model composer product consists of four composer tools: the unit, entity, behavior and environment. The components are briefly described within the model composer product (Wittman Jr & Harrison, 2001).

- Unit Composer

The Unit Composer provides the capability to construct hierarchical military units (or organizations) from other unit constructs and entities. Information describing the new unit can then be entered within the unit composer tool. The unit composer can also allow behaviors such as search to be bound to specific units (Wittman Jr & Harrison, 2001).

● Entity Composer

The Entity Composer provides the capability to construct battlespace entities like tanks from supporting constructs such as hulls, tracks, turrets, sensor, guns, etc. Information describing the new entity can then be entered within the entity composer tool. The entity composer will also allow behaviors including direct fire controller, operations, intelligence, and supply, and physical models such as sensors (e.g. eyeball, FLIR, etc.), weapons, mobility, and vulnerability to be bound to specific entities (Wittman Jr & Harrison, 2001).

● Behavior Composer

The Behavior Composer provides the capability to build complex behaviors using a flowchart graphical language from other primitive behavior types. Primitive behaviors provide chunks of functionality from which more complex behavior models are built and are parameterized with inputs, and may have outputs. Composite behaviors represent tasks and missions and are composed of primitive and other composite behaviors. Complex behaviors, along with their relevant metadata, will be specified in an XML based behavior specification language. Information describing the new behavior can then be saved within the behavior composer tool (Wittman Jr & Harrison, 2001).

● Environment Composer

The Environment Composer provides the user the capability to compose the synthetic environment to include, but not limited to, geographic location, terrain representation and resolution, feature representation and resolution, atmospheric effects representation and resolution, bathymetric representation and resolution, etc (Wittman Jr & Harrison, 2001).

5.2.1.2.2 Simulation Generator Product

The Simulation Generator Product provides the selection of the appropriate terrain and environmental information, forces, factional relationships, non-combatant organizations, data collection information and other elements necessary to capture the requirements of the scenario at execution. The selection process is supported by the examination of metadata describing each element. The Generator uses the XML Military Scenario Specification created by the MSDE component as a basis for extension. The Simulation Generator supports association of synthetic entities with map based control measures and temporal order execution sequences. The Simulation Scenario Specification is stored in an XML based format for further processing by the Technical Manager Product. The Components within this Product are briefly described below (Wittman Jr & Harrison, 2001).

● Military Scenario Generator: The Military Scenario Generator provides the GUI-based mechanism for the selection of appropriate forces, factional relationships, non-combatant organizations, and other elements necessary to capture the requirements of the scenario at execution. It updates the Simulation Scenario Specification with this additional data (Wittman Jr & Harrison, 2001).

● Environment Database Generator: The Environment Database Generator Component provides the GUI based mechanism for the selection of appropriate terrain and environmental data necessary to capture the requirements of the scenario at execution. It updates the Simulation Scenario Specification with this additional data (Wittman Jr & Harrison, 2001).

● Data Collection Tool: The Data Collection Tool will allow the user to identify the data items of interest for collection during simulation execution. It updates the simulation scenario specification with this additional data (Wittman Jr & Harrison, 2001).

5.2.1.2.3 Repository Manager Product

Repository Manager Product accommodates all CS-SSAF data and information. The users may utilize and manage the storage of data.

● Data Management Tool

Data Management Tool provides mechanisms to access, review, modify, archive, and analyze data within the Repository Manager.

● Information Metadata Tool

Information Metadata Tool performs a management of the metadata which is stored in the repository.

5.2.1.2.4 *Review Analysis* Product

The Review Analysis Product shall support mining of collected data to construct Measures of Effectiveness (MOEs)/Measures of Performance (MOPs) and analytical charts and graphs as well as allowing data export to COTS Office Automation and analytical review tools (Wittman Jr & Harrison, 2001).

- After Action Review (AAR)

After Action Review (AAR) is the primary method for delivering feedback after individual or unit training exercises (Morrison & Meliza, 1999). The AAR product supports graphical review such as the snapshots of the simulated scenario, analysis and presentation of all data collected during the simulation execution. The toolset shall support mining of the collected data to construct Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and analytical charts (Wittman Jr & Harrison, 2001). Figure 93 illustrates the OneSAF AAR Architecture (Morse, 2010).

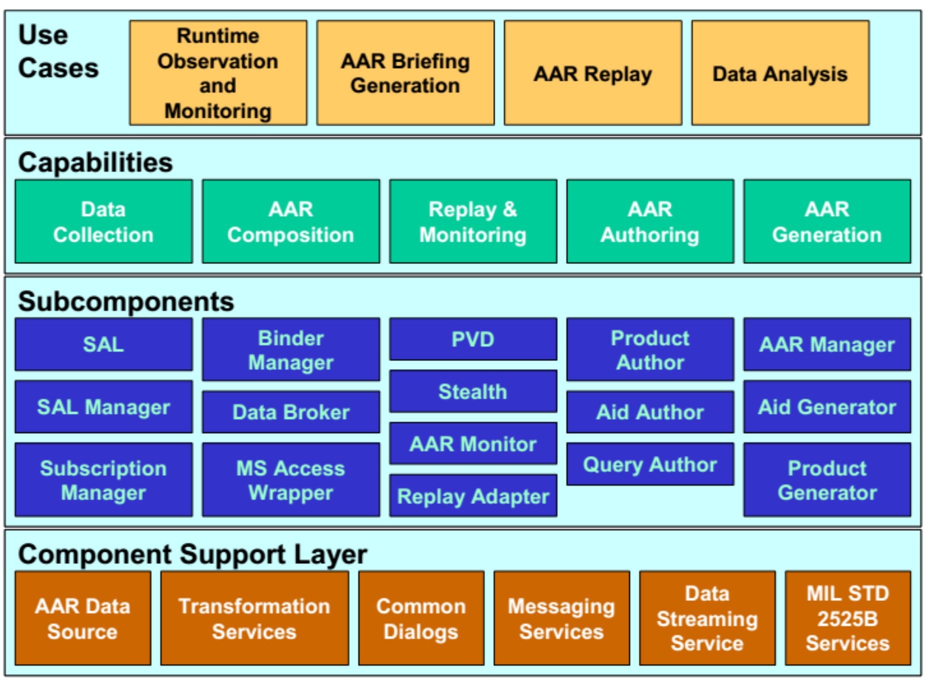


Figure 93: OneSAF AAR Architecture

● Annotator Tool

The Annotator Tool will provide an observer/controller or other remote user the ability to record electronic form based data entry regarding the simulation event to support AAR and Analysis activities. It is envisioned that this will be implemented in a Personal Digital Assistant (PDA)-based application (Wittman Jr & Harrison, 2001).

● Battle Damage Assessment (BDA)

The estimate of damage results is calculated from the application of lethal or nonlethal military force. BDA is composed of physical damage assessment, functional damage assessment, and target system assessment.

5.2.1.2.5 Common Command, Control, Communications, Computers and Intelligence (C4I) Adapter Product

The C4I Adapter is a software tool that provides bi-directional translation, connection, routing and control and monitoring of information flowing between real-world battle-command (BC) devices and Constructive simulation system. The ultimate objective of common C4I product is to integrate the C4I Adapter within other programs with similar C4I interface requirements. The components within this product are briefly described below.

● Translation Service

Translation Services will provide two way translation services that translate internal simulation system formats to C4I formats and vice versa (Wittman Jr & Harrison, 2001). It will support translation of data between the various formats.

● Connect Service

Connect Services will provide a mechanism to connect the Adapter to specific C4I systems using inherent C4I protocols and physical connection mechanisms. These may include but are not limited to serial communication lines, Ethernet, wireless communications, etc (Wittman Jr & Harrison, 2001).

● Monitor & Control Service

Monitor and Control Services will provide mechanisms to monitor and control the C4I adapter settings as well as manage, control, or modify the data flowing between the C4I system and simulation system (Wittman Jr & Harrison, 2001).

5.2.1.2.6 Distributed Simulation Extender Product

Distributed Simulation Extender Product creates and manages simulation environment (or federation) for LVC simulation. It provides a gateway tool to interconnect simulation systems which use different SSAs.

● Simulation Environment Development/Management Tool

Simulation Environment Development Tool provides a GUI-based mechanism for supporting the HLA, DIS, TENA and CTIA simulation environment development process. This tool shall support SOM to FOM mapping in support of HLA federation execution.

● Network Loader Tool

Network Loader Tool provides a GUI-based mechanism to assess network performance and capacity to support a simulation system execution (Wittman Jr & Harrison, 2001).

● Gateway Tool

Gateway Tool supports the interoperability between different SSAs such as HLA 1.3, HLA 1516, HLA 1516e, DIS, TENA and CTIA.

5.2.1.2.7 Visualization Product

Visualization Product supports realistic 3D view of the Virtual battlefield.

● Stealth Tool

Stealth Tool displays a realistic, 3D representation of the virtual battle space. User can view the Virtual world from inside a simulated moving vehicle, or place the eye-point at another moving or stationary location (MÄ K, 2011). It also provides flexible eye-point control, including the ability to attach to different SAF and virtual simulation entities. User can switch rapidly among several predefined viewpoints using the stealth tool during the simulation execution.

● Visual Image Generator (IG)

Visual Image Generator (IG) provides realistic 3D scenes of the Virtual simulation environment.

5.2.1.2.8 *CGF* Product

CGF product is a collection where common CGF models are stored. The CGFs are used in Live, Virtual and/or Constructive simulation systems.

● Unit Model

Unit Models are comprised of military organizational or unit models. The unit is defined as a component of a military, paramilitary, quasi-military such as guerilla or terrorist cell, etc., governmental or other organizational hierarchy. Traditional military units are organized by echelon such as soldier, team/crew, squad, platoon, company, battalion, regiment and brigade. The Unit Models provide the runtime representation of the Units identified within the Simulation Scenario Specification (Wittman Jr & Harrison, 2001).

- Entity Model

An entity may be a life form such as human and animal or a platform such as tank and helicopter. The Entity Models also provide the runtime representation of the Entities identified within the Simulation Scenario Specification (Wittman Jr & Harrison, 2001).

- Behavior Model

Behavior Models provide the runtime modeling of the cognitive aspect of Units and Entities and utilize the XML based behaviors that have been composed for each of the scenario's units and entities (Wittman Jr & Harrison, 2001).

- Physical Model

Physical models provide the mathematical representation of combat systems and their interactions with the environment and other entities (Wittman Jr & Harrison, 2001).

- Environmental Model

Environmental Model is comprised of environmental models, both dynamic and static. It provides the GUI based mechanism for the selection of appropriate terrain and environmental data necessary to capture the requirements of the scenario at execution (Wittman Jr & Harrison, 2001).

5.2.1.3 Component Support Layer

Components in the Component Support Layer directly support the components which support the product layer.

5.2.1.3.1 GUI Service

CS-SSAF provides a GUI-based mechanism to manage, control, or modify the terrain and environmental information, forces, factional relationships, non-combatant organizations, data collection information and other elements (Wittman Jr & Harrison, 2001).

5.2.1.3.2 Composition Service

Composition Service is well supported so that the components are assembled.

5.2.1.3.3 Data Collection Service

Data Collection Service provides the services to collect and store all of the data identified for supporting AAR and BDA.

5.2.1.3.4 Simulation Service

Simulation services are services to perform basic functions such as simulation time progresses, event management, and random number generation during simulation execution.

5.2.1.3.5 Modeling Service

Modeling Service provides services necessary to configure the new models for LVC simulation.

5.2.1.3.6 Environment Service

Environment Service supports dynamic environmental changes that occur in the simulation system.

5.2.1.3.7 Plan View Display (PVD)

Visualization varies across simulation systems, from 2D considering a unit level such as a brigade to 3D considering levels of an entity or unit such as a soldier or squad level (Tolk, 2012).

Plan View Display (PVD) provides a 2D plan view display. PVD views can show raster graphic maps or top-down views of the terrain database. User can find all the functionality to create and run a scenario in the 2D plan view (MÄ K). PVD displays situational information about simulation entities on the map. The VR-Forces simulation system has the capability of modeling AI based automated entity behaviors.

5.2.1.3.8 Visual Data Service

Visual Data Service provides GUI-mechanisms to monitor, manage, and/or modify the data. It provides positional awareness via a 3D viewer and a 2D map display of the battlefield.

5.2.1.3.9 Simulation Execution Service

The simulation event is executed.

5.2.1.3.10 Report Service

The Report Service allows any application that runs within the CS-SSAF to generate reports based on events that have occurred in the past.

5.2.1.4 Repository Layer

Repository Layer accommodates all simulation system data and information. The repository must accommodate, at a minimum, the following types of data: system and software documentation, system and software source code and executable code, system and software product configuration data and change history, any metadata necessary to support simulation composition activities, scenario data, simulation execution data, simulation execution performance metrics, results of analysis performed on simulation data, after action review data, etc (Wittman Jr & Harrison, 2001).

5.2.1.4.1 KI/KC/CM Repository

As mentioned above, we proposed DCMF when M&S community develops the conceptual modeling. The DCMF process is comprised in four main parts; Knowledge Acquisition (KA), Knowledge Representation (KR), Knowledge Modeling (KM) and Knowledge Use (KU) phase. KA/KR/KM/KU repository. These phases generate three of the most important outputs; Knowledge Instances (KI), Knowledge Components (KC) and Conceptual Models (CM). These outputs are stored for future use and reuse.

5.2.1.4.2 Environment Repository

Environment Repository stores Master Database (MDB) to support the correlated terrain database. A master database is populated from a union of multiple authoritative data sources.

5.2.1.4.3 Simulation Output Repository

Simulation Output Repository is a repository that stores all the output generated during the simulation LVC. This repository stores the metadata, data logs, and system evaluation, checkpoints and reply file.

5.2.1.4.4 Software Repository

Software Repository stores software documentation, software source code and executable code and software product configuration data and change history (Wittman Jr & Harrison, 2001).

5.2.1.4.5 Military Scenario Repository

Military Scenario Repository stores all the necessary military scenarios for the LVC simulation.

5.2.1.4.6 Parameter Repository

Parameter Repository stores all the parameters to initialize or to run the LVC simulation.

5.2.1.5 Service Layer

The CS-SSAF services are a set of common software service interfaces that provides the framework or infrastructure on which CS-SSAF common components are built.

5.2.1.5.1 Time Service

Time Service provides time synchronization.

5.2.1.5.2 Monitor Service

Monitor Service is a service that monitors the load on the computer and the network.

5.2.1.5.3 Messaging Service

This service supports communication through the exchange of messages between components constituting the LVC system.

5.2.1.5.4 Coordinate Service

This service provides the coordinate conversion library between LVC simulation systems using different coordinate system.

5.2.1.5.5 Interoperation Service

Interoperation Service provides protocol translation conversion between distributed LVC simulation systems using different SSAs.

5.2.1.6 Middleware Service

Middleware service represents SSAs, such as HLA, DIS, TENA and CTIA.

5.2.1.6.1 High Level Architecture (HLA) /Runtime Infrastructure (RTI)

HLA was selected due to its high usage within the M&S communities. The Real-time Platform- level Reference (RPR) FOM is considered as a common standard for the CS-SSAF HLA specification.

5.2.1.6.2 Distributed System (DIS)

DIS was selected to support Virtual simulation systems.

5.2.1.6.3 World Wide Web (WWW)

WWW service is provided.

5.2.1.6.4 TENA

Because the RPR FOM and the DIS cannot support Live training, TENA was selected to support integrating Live assets in the test-range setting.

5.2.1.6.5 CTIA

CTIA also was selected to support interconnecting Live assets. CTIA can promote commonality among the instrumented ranges and home stations.

5.2.1.7 Platform Layer

The platform layer shows the host hardware, operating systems, and network technology supported by the CS-SSAF. These are usually commercial off-the-shelf (COTS) or open source.

5.2.2 Common Standard-Live Simulation System Architecture Framework (CS-LSSAF)

We developed the Common Standard-Live Simulation System Architecture Framework (CS-LSSAF) from CS-SSAF. We identified four additional products and twenty-four components for CS-LSSAF. Figure 94 shows CS-LSSAF, and the purple font color indicates unique products and components for CS-LSSAF.

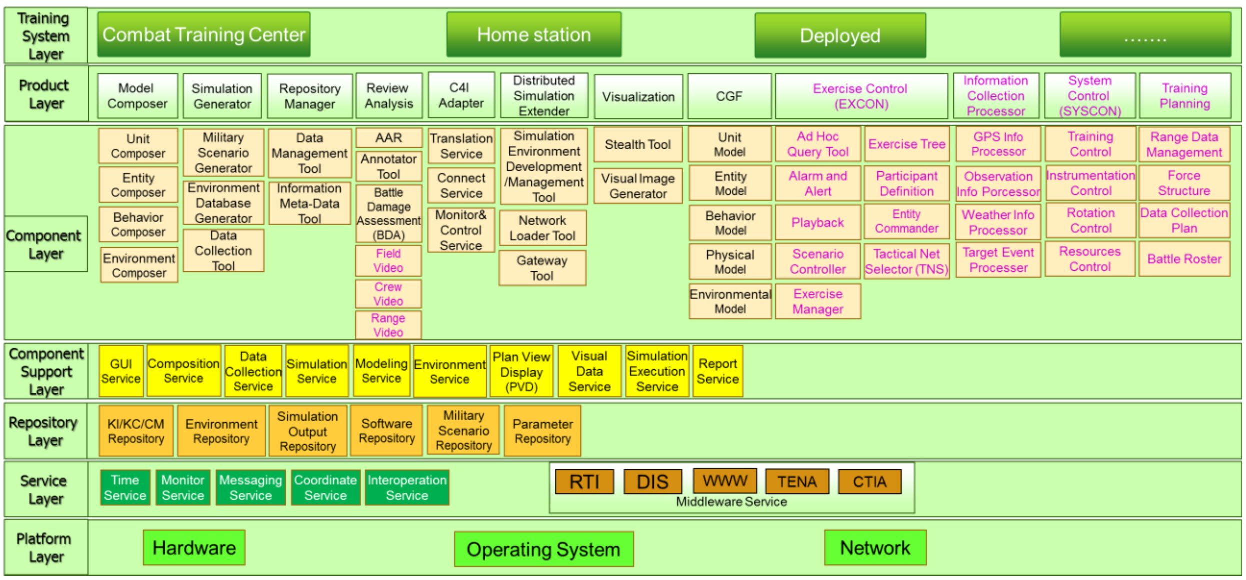


Figure 94: CS-LSSAF

The following subsections describe the each of unique components of CS-LSSAF by referring to the Live Training Product Line (LT2) Overview Briefing (*CTIA Live Training Product Line (LT2) Overview Briefing*, 2006).

5.2.2.1 Review Analysis Product

This section explains additional components of the Review Analysis Product for CS-LSSAF.

5.2.2.1.1 Field Video

The Field Video provides the capability to command camera mounts which points to specified locations via presets that have been programmed into the camera mounts.

5.2.2.1.2 Crew Video

The Crew Video provides the capability for a user to assign a player unit's video cameras to one of the available channels.

5.2.2.1.3 Range Video

The Range Video records range.

5.2.2.2 Exercise Control (EXCON) Product

This section covers the Exercise Control.

5.2.2.2.1 Ad Hoc Query Tool

The CS-LSSAF Ad Hoc Query is a component designed for making CS-LSSAF framework widgets for the purpose of creating the exercise report.

5.2.2.2.2 Alarm and Alert

The Alarms and Alerts Component (AAC) is a component developed for analyzing, publishing, and detecting Alarm and Event Subscriptions.

5.2.2.2.3 Playback

The Playback component enables users to replay activities that occurred during a training exercise.

5.2.2.2.4 Scenario Controller

The Scenario Controller is responsible for commanding and controlling physical range assets during an exercise and provides all of the logic involved in executing an exercise.

5.2.2.2.5 Exercise Manager

The Exercise Manager provides configuration, control, and views of the exercise instantiation in the system.

5.2.2.2.6 Exercise Tree

The Exercise Tree is a component used for viewing and editing relevant objects to the training audience.

5.2.2.2.7 Participant Definition

The purpose of the Participant Definition Tool (PDT) component is to allow the end user to create/edit participant entities as part of an exercise.

5.2.2.2.8 Entity Commander

The Entity Commander component provides a set of commands available to update controlled entities

5.2.2.2.9 Tactical Net Selector (TNS)

The Tactical Net Selector (TNS) component provides the Tactical Analysis and Feedback (TAF) workstations with the ability to monitor radio traffic and play back recorded radio traffic.

5.2.2.3 Information Collection Processor Product

This section covers the information collection processor product.

5.2.2.3.1 Global Positioning System (GPS) Information Processor

GPS Information Processor reads the data from the GPS via a network.

5.3.2.3.2 Observation Information Processor

Observation Information Processor component is developed for creating, viewing, editing and deleting observation.

5.2.2.3.3 Weather Information Processor

Weather Information Processor reads messages from the weather station and converts the message into CS-LSSAF state messages.

5.2.2.3.4 Target Event Processor

Target Event Processor represents the current state of the targets as related to the CTIA Exercise

5.2.2.4 System Control (SYSCON)

This section covers the CS-LSSAF system control components.

5.2.2.4.1 Training Control

Training Control component is a tool that is used to control the whole training and monitor the training situation.

5.2.2.4.2 Instrumentation Control (ISC)

Instrumentation Control (ISC) provides the ability to monitor the status of various instrumentation devices, such as Player Units (PUs), and sends them commands.

5.2.2.4.3 Rotation Control

Rotation Control is a component to define a rotation, prepare a rotation, run a rotation, and manage a rotation.

5.2.2.4.4 Resources Control

Manager component is responsible for assisting the user with the allocation and management of training resources for instrumented, live collective training exercises.

5.2.2.5 Training Planning

This section covers the CS-LSSAF planning components.

5.2.2.5.1 Range Data Management

The Range Data Editor component can be used to manage the allocation of range assets (e.g., targets, target lifters, cameras, etc.) to a specific range and information associated with their use at that range.

5.2.2.5.2 Force Structure

Force Structure is a CTIA-compliant component that is responsible for creating and editing force structures.

5.2.2.5.3 Data Collection Plan

The Data Collection Plan (DCP) component provides the ability for a database administrator to easily manage and manipulate data within a DCP database.

5.2.2.5.4 Battle Roster

The Battle Roster is used during exercise planning to import battle roster data into the exercise database. A typical battle roster contains a list of participants that are being trained in an exercise.

5.2.3 Common Standard-Virtual Simulation System Architecture Framework (CS-VSSAF)

This section describes the common standard-virtual simulation architecture framework (CS-VSSAF).

In the Training System Layer, the Virtual simulator domain can be divided into four major classifications: (a) individual, (b) crew, (c) collective and (d) combined arms. The CS-VSSAF domain will include all the classifications defined above and shall support the Army, Navy and Air-Force.

In order to develop CS-VSSAF, first, we gathered the needs of the Virtual domain to determine common standard Virtual components requirements through the requirement analysis. Second, we determined the best-fit reuse products and components to meet common standard Virtual component requirements through reuse analysis. Then, we designed the common standard Virtual components within the Common Standard-Live Simulation System Architecture Framework (CS-VSSAF) from CS-SSAF. We identified two additional products in product layer, fifteen components in component layer, five components in component support layer, and five components in repository layer for CS-VSSAF. Figure 95 shows CS-VSSAF, and the red color font indicates unique products and components for CS-VSSAF.

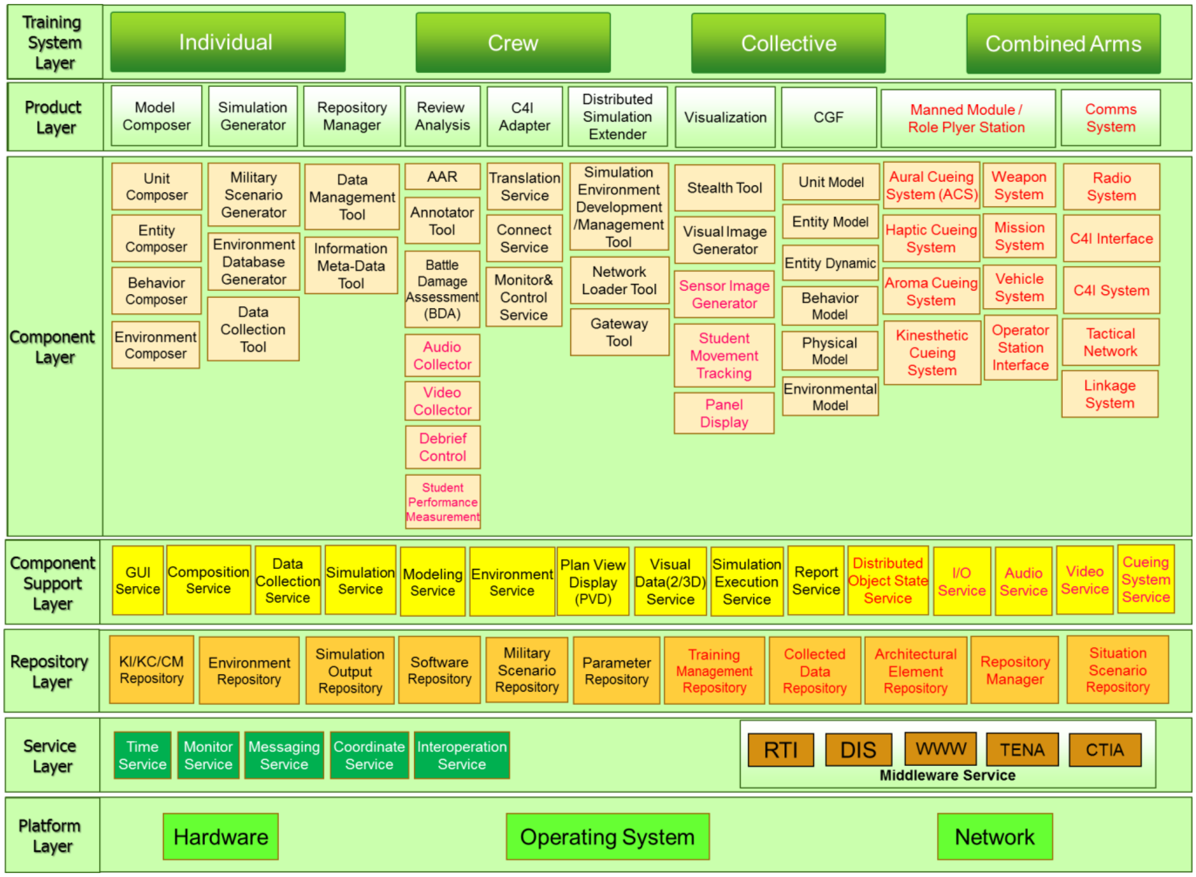


Figure 95: CS-VSSAF

The following subsections describe the each of unique components of CS-VSSAF.

5.2.3.1 Review Analysis Product

This section covers Review Analysis Product of CS-VSSAF.

<u>5.2.3.1.1 Audio Collector</u>

Audio Collector is a component that collects voice information generated from a trainee, or trainer and simulator during the simulator execution.

5.2.3.1.2 Video Collector

Video Collector is a component that collects video information generated during the simulator execution.

5.2.3.1.3 Debrief Control

Debrief Control support debrief presentation.

5.2.3.1.4 Student Performance Measurement

Student Performance Measurement is a component for measuring and evaluating the training performance of the students or trainees.

5.2.3.2 Visualization Product

Visualization products are designed to meet user needs for visualizing the simulated world.

5.2.3.2.1 Sensor Image Generator

Sensor Image Generator supports images that sensors such as radar and Forward Looking Infrared Radar (FLIR)

5.2.3.2.2 Student Movement Tracking

Student Movement Tracking supports to track the movement of the trainee to display the virtual reality.

5.2.3.2.3 Panel Display

Panel Display supports LCD touch panel displays depicting a graphical representation such as the flight deck including the overhead panel, dual sided displays, upper and lower displays and the pedestal, thrust levers, flap lever, speed brake lever and other components. Trainees can truly be immersed in the simulation through panel display.

5.2.3.3 Manned Module / Role Plyer Station Product

This section covers Manned Module and Role Player Station Product. Virtual reality at present primarily involves the sense of vision, however several Virtual simulation systems such as tank and helicopter operate in a multisensory world.

5.2.3.3.1 Aural Cueing System (ACS)

This component reproduces the exact sound and noise from the actual equipment during simulation execution.

5.2.3.3.2 Haptic Cueing System

It is very important to complement the visual information through the sense of touch. Haptic Cueing System provides the sense of touch to improve the human machine interfaces.

5.2.3.3.3 Aroma Cueing System

Aroma Cueing System creates a variety of aroma based on battlefield environments.

5.2.3.3.4 Kinesthetic Cueing System

Kinesthetic Cueing System provides trainees with motion perception during simulation execution. It is most important when driving a simulator such as tank.

5.2.3.3.5 Weapon System

Weapon System defines the model of a weapon system that Virtual simulator simulates.

5.2.3.3.6 Vehicle System

Vehicle System defines the model of a vehicle system such as a tank and truck that Virtual simulator simulates.

5.2.3.3.7 Operator Station Interface

Operator Station Interface provides the operator with interface. Operator Station Interface receives the input data and shows output data.

5.2.3.4 Communications System Product

This covers the Communications System Product.

5.2.3.4.1 Radio System

Radio System provides trainee with realistic radio communications within Virtual environment. It provides the replication of real world effects such as radio signal degradation based on terrain, radio types and environmental noise to increase the realism.

5.2.3.4.2 C4I Interface

C4I Interface provides interface between live C4I system and the Virtual simulator.

5.2.3.4.3 C4I System

C4I System allows the Virtual simulator to get information from live C4I.

5.2.3.4.4 Tactical Network

Tactical Network enables communication between military communication platforms and the Virtual simulator.

5.2.3.4.5 Linkage System

Linkage System enables linkage to other simulation systems, such as Live training system and Virtual simulators, Constructive simulation, Web and commercial engineering tools.

This concludes the development of Common Standard-Virtual Components (CS-VC) that will reduce redundancy, increase realism, and facilitate an integrated Live, Virtual and Constructive training environment.

5.2.4 Common Standard-Constructive Simulation System Architecture Framework (CS-CSSAF).

Finally, this section describes Common Standard-Constructive Simulation Systems Architecture Framework (CS-CSSAF). We identified one additional component from the CS-SSAF, an Icon Tool component as shown in Figure 96. The blue color font indicates the Icon Tool component.

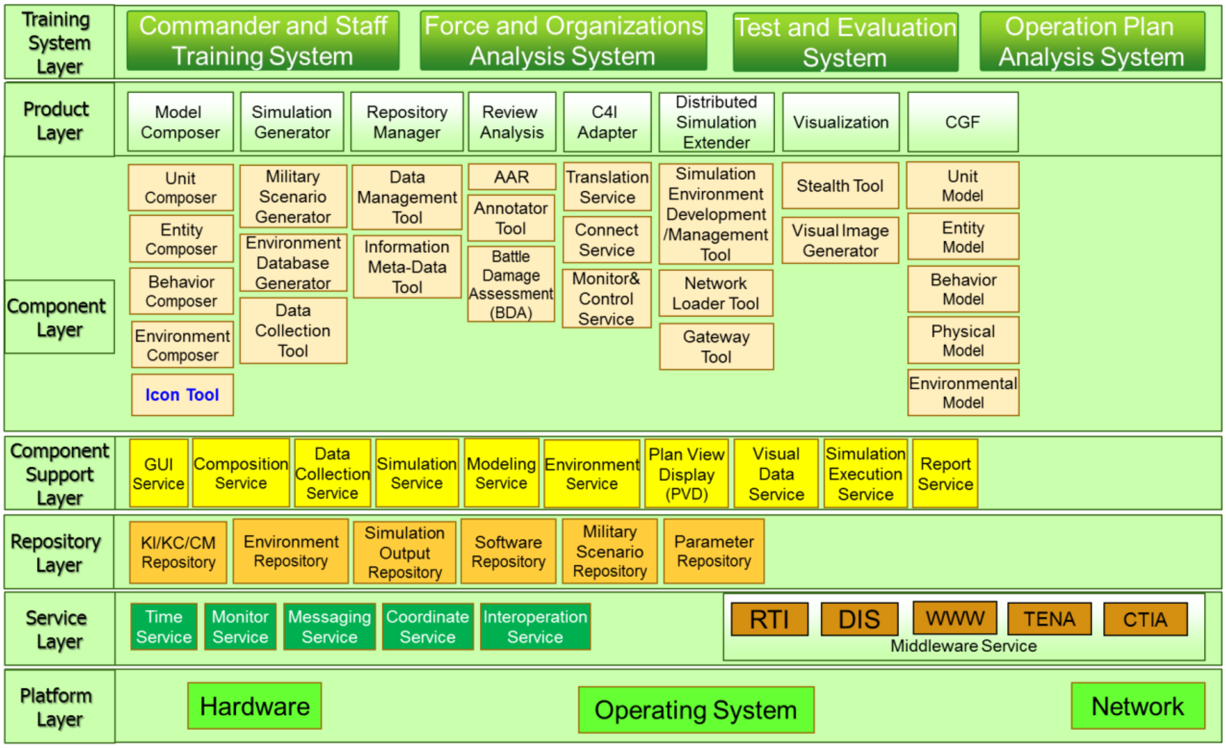


Figure 96: CS-CSSAF

5.2.4.1 Model Composer

This section covers the Model composer of CS-CSSAF.

5.2.4.1.1 Icon Tool

2D icons are specified by the Military Symbol Icon Visualizer in the entity definition. If some M&S community has a plan to do joint operations with the U.S. military, we recommend the MIL-STD 2525B icons. 2D icons can display the entity's: Name, Orientation, Velocity, Acceleration, Location and Heading Indicator. Figure 97 illustrates basic icon shapes for friendly forces.

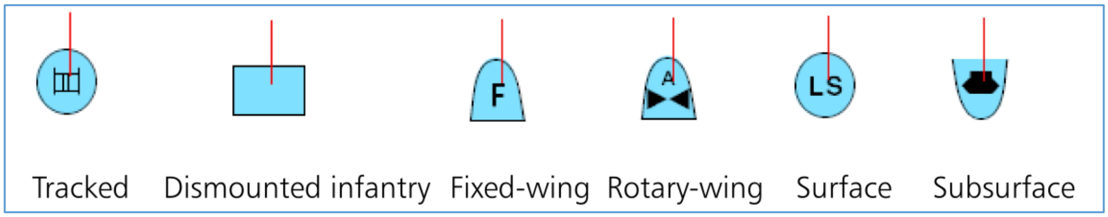


Figure 97: MIL-STD-252B Icons

5.3 Recommended Action 3: Common Standard Correlated Terrain Database (CS-CTDB)

Recommended action 3, common standard correlated terrain database (CS-CTDB) is recommended to realize lesson learned 7, need for a common correlated terrain database. This section covers the CS-CTDB.

The use of correlated TDB in the two or more systems was absolutely critical to the successful interoperation of multiple LVC simulation systems. The objective of the recommended action 3 is to provides a common correlated terrain database, resulting a fair fight environment between multiple simulation systems

Therefore, I proposed a parallel development strategy for developing CS-CTDB. Strategy 1 is to use a legacy simulation system. Strategy 2 is to develop the CS-CTDB with reference to Standard/Rapid Terrain Database Generation Capability (STDGC). The following sections describe each of the development strategy in detail.

5.3.1 Strategy 1: Reuse Legacy Simulation System

Strategy 1 is to use a legacy simulation system. Usually, M&S developing communities (or countries) have a Representative Constructive Simulation (RCS) and Representative Virtual Tactical Training Simulator (RVTTS). The primary goal of the first strategy is to integrate the Constructive simulation and Virtual simulators. Strategy 1 consists of four phases: (a) Choice of RCS and RVTTS, (b) RCS-ERC Development, (c) Attainment of Interoperability, and (d) RCS-ERC Integration into RVTTS.

The following sections describe each of the steps in detail.

5.3.1.1 Phase 1: Choice of RCS and RVTTS

M&S developing communities (or countries) have to choose a Representative Constructive Simulation (RCS) to integrate Representative Virtual Tactical Training Simulator (RVTTS). The RCS and RVTTS must be the simulation systems that are likely to be developed. For example, South Korea can select the AddSIM as a RCS. The AddSIM is currently in the development process, but many capabilities will be added and reinforced.

5.3.1.2 Phase 2: RCS-ERC Development

The RCS, M&S community chooses, shall have the Environmental Runtime Component (ERC) capability such as OneSAF's ERC. If the RCS does not have the ERC, we can have two alternatives. The alternatives are: (a) development of ERC and (b) reuse of an existing SNE software.

First of all, I describe what the ERC is. OneSAF ERC provides urban terrain features and ultra-high resolution buildings facilitating training in the contemporary operating environment.

In addition, OneSAF ERC provides the static environmental representation (Land, Sea, Air, Space): coordinate services, data models (shared), runtime compilers, and environmental effect models such as NBC, smoke, dust, dynamic terrain/atmosphere, etc. (Logsdon & Wittman, 2007).

Next, two alternatives including the ERC are as follows:

● Alternative 1: M&S community has to develop the RCS-ERC such as OneSAF-ERC.

● Alternative 2: If M&S communities have an existing Synthetic Natural Environment (SNE) software, they can reuse and develop it in order to minimize life cycle maintenance cost and software development for the RCS.

5.3.1.3 Phase 3: Attainment of Interoperability

If the M&S community finished the Phase 2, the community will equip the RCS-ERC. Then, M&S community has to facilitate the attainment of interoperability requirements between the RCS-ERC and the legacy terrain database of the VRTTS.

5.3.1.4 Phase 4: RCS-ERC Integration into RVTTS

In Phase 4, RCS-ERC will be integrated with RVTTS. Through the integration, the legacy terrain format and environmental services in the RVTTS are replaced with the RCS-ERC. Figure 98 shows the integration process between RCS-ERC and RVTTS.

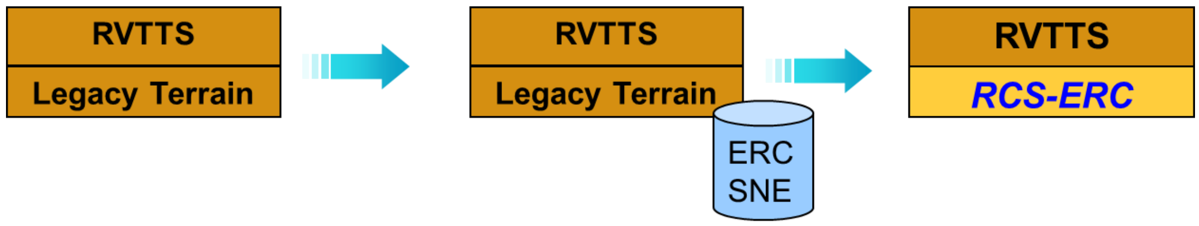


Figure 98: Integration Process between RCS-ERC and RVTTS

M&S community can get benefits from the integration between RCS-ERC and RVTTS.

First, the interoperability will be enhanced between RCS and RVTTS. Second, the RCS will be used to extend the simulation system capability. Third, the common environmental service will be implemented between RCS and RVTTS. Fourth, the RVTTS's existing legacy terrain database and service will be retired. Lastly, the RVTTS will benefit from embedded RCS capability such as RCS-ERC.

In conclusion, through the Strategy 1, we can achieve the replacement of the legacy terrain formats and environmental services in representative with RCS-ERC. Strategy 1 will put the RCS and RVTTS on the same TDB and enhance the correlation and interoperability between the RCS and RVTTS.

5.3.2 Strategy 2: Develop CS-CTDB Generation System

The second strategy is the development of the new CS-CTDB generation system in M&S developing community (or country) for LVC simulation

In Section 2.7.5.2.4, we reviewed the U.S. Army's SE Core Standard/Rapid Terrain Generation Capability (STDGC). The SE Core's primary mission is to rapidly generate correlated simulation system terrain databases. However, the SE Core program has been focused on supporting the Virtual domain.

The CS-CTDB generation system will produce correlated industry standard and runtime terrain databases for gaming systems as well as LVC simulation systems using standards and standardization within 72 hours as shown in Figure 99.

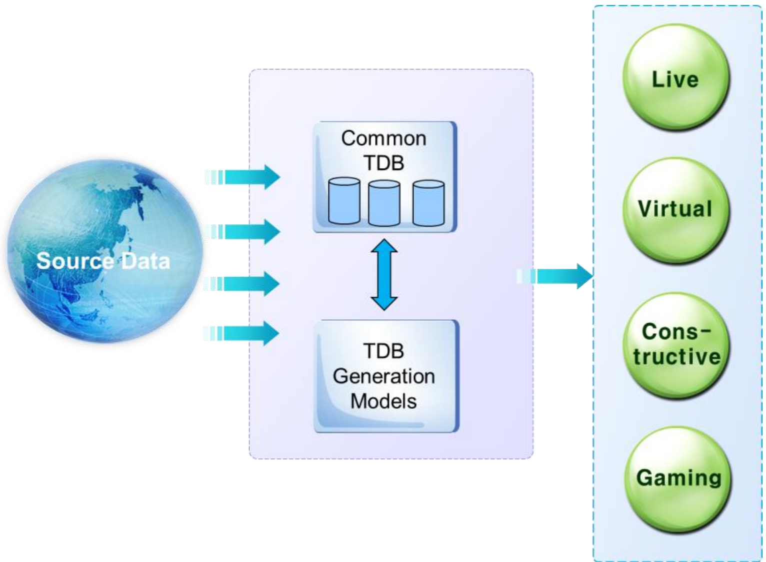


Figure 99: Common Standard-Correlated Terrain Database (CS-CTDB)

5.3.2.1 Phase 1: Construction Master Terrain Database (TDB) Generation Centers

M&S developing community will need a Master Terrain Database Generation (M-TDB) Centers for supporting the CS-CTDBs. The M-TDB is the central repository for the creation of correlated terrain databases used to train, mission plan, or mission rehearsal in the LVC domains. This M-TDB is integrated to standards and readied for consumption in specific geospatial data sets and runtime formats used by many training systems.

M&S developing community may have two strategies for constructing the master TDB generation center manage the master TDB.

Strategy 1 is to improve and expand the existing terrain information facilities. For example, Korea Defense Geospatial Intelligence Agency in South Korea will be able to perform the role as a common database generation center. Strategy 2 is to construct the new common database generation center.

The founded master TDB generation center should maintain other terrain information agencies to request the required terrain data format directly. For creating CS-CTDBs, typical raw source data will be collected from many source providers, including National Geospatial - Intelligence Agency (NGA), Commercial, Joint services, Agencies, Governments, and Countries. Figure 100 shows the cooperation between the master terrain database generation centers.

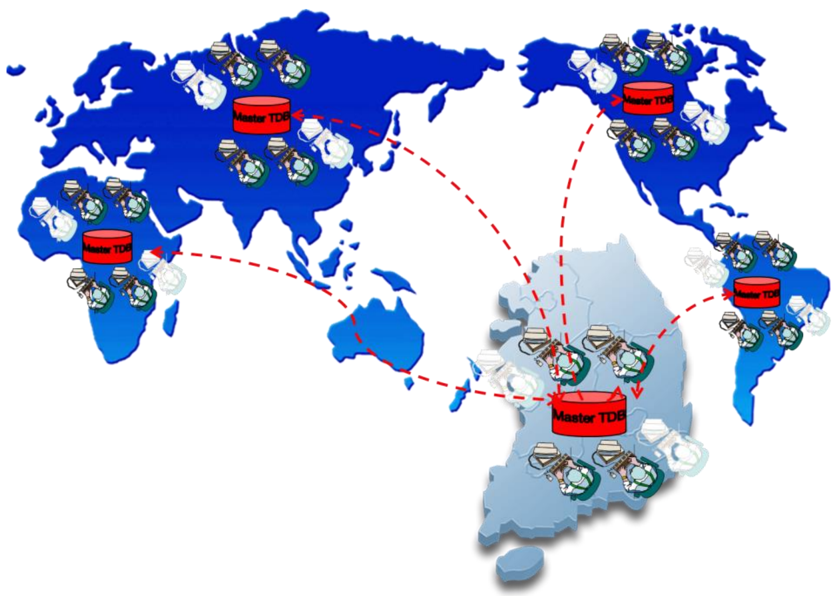


Figure 100: Master Terrain Database Generation (TDB) Centers

5.3.2.2 Phase 2: Source Data Management

These source providers may collect the raw source data from manned ground vehicles, battle field sensors, unmanned ground vehicles, unmanned air vehicles, and satellite pictures as well as other future platform sensors and sources of intelligence (Graniela & Proctor, 2012).

The collected raw source must be managed systematically in the form of source interchange formats. The main data type that the CS-CTDB uses may include:

● Imagery: CIB, Buckeye, and JPEG

● Vector: VMAP, Urban Tactical Planner, NAVTEQ, DAFIF, and Shape Files

● Elevation: DTED and LIDAR

● Models: Site Photos, Building diagrams and CAD

From the data type above, the CS-CTDB will support some of the format, including DTED, Shape Files, OTF, CTDB, Open Flight, VBS2, SEDRIS Transmittal Format, etc.

In conclusion, we expect that a CS-CTDB generation system will produce heterogeneous target formats more rapidly and efficiently, while maintaining correlation with each other for future heterogeneous network-centric simulation systems.

5.4 Recommended Action 4: Advanced Interoperability Technology

Recommended action 4, advanced interoperability technology is recommended to realize: (a) lesson learned 5, need for multiple SSAs compliancy, (b) lesson learned 6, need for a scalability of simulation systems, (c) lesson learned 8, need for a new C-SSA, and (d) lesson learned 12, need for a general bridging tool. This section covers the advanced interoperability technology.

Substantive interoperability between LVC simulation systems is essential to providing the highest quality warfighter training. Simulation systems require compliance of the SSA in order to improve interoperability. As described above, a number of SSAs such as ALSP, DIS, HLA, CTIA and TENA are developed to meet the interoperability needs. Although HLA is an

IEEE (1516) and NATO standard developed for simulation systems and widely used all around the world, it is not compatible with other SSAs. Therefore, the simulation systems shall ease integration to other simulation systems (or federates) based on different SSAs. An Interoperability Manager (IM) of simulation systems must be designed to interoperate with the different SSAs in order to work together effectively in the M&S environment.

5.4.1 Policy Establishment on Multiple Standard Simulation Architecture (SSAs)

In order to interoperate and reuse developed M&S resource, we need to apply SSAs such as HLA, DIS, TENA, CTIA and so on. However, M&S developing communities (or countries) did not consider the purpose and application area well, and just tried to apply several SSAs to their M&S systems.

Therefore, M&S developing communities need to establish the policy about several SSAs. Table 25 summarizes and compares the main technology between M&S developing communities and the U.S. Army in M&S domain.

Table 25: Comparison about Main Technology in M&S Domain

Items	M&S Developing Communities	U.S. Army
Standard Simulation Architectures (SSAs)	HLA, DIS	HLA, DIS, TENA, CTIA
Simulation Environment (or Federation)	• Virtual – Constructive • Live - Constructive	Live-Virtual-Constructive
Common Simulation Architecture Framework	None	• Live: LT2-FTS • Virtual: SE-Core • Constructive: JLCCTC
Synthetic Environment (SE)	None	SE-Core
Interoperability Tech. Level	HLA/RTI	HLA, TENA, DIS, CTIA → LVC-IA Development
Goal	LVC Simulation	LVC Simulation: LVC-IA

First, in the SSA domain, while most M&S developing communities use HLA and DIS, the U.S. Army, in addition, uses TENA and CTIA for Live domain.

Second, in the simulation environment domain, while M&S developing communities mainly interconnect between Virtual and Constrictive (VC), or between Live and Constructive (LC), the U.S. Army implemented the LVC simulation.

Third, in the common simulation architecture framework domain, most M&S developing communities do not have a common simulation architecture framework, and developed several M&S systems as needed without a long term master plan. However, the U.S. Army has been applying the PLAF concept in developing M&S systems.

Fourth, in the synthetic environment domain, while most M&S developing communities do not have the common correlated TDBs and have specific TDB for only their own system, the U.S. Army developed SE-Core and has been applying to all M&S systems.

Finally, in the interoperability technology domain, while most M&S developing communities have been developing the M&S technologies only related to HLA/RTI, the U.S. Army have developed M&S technologies related to several SSAs and interconnected them through LVC-IA.

The goal of both M&S developing communities and the U.S. Army is to achieve an LVC simulation and the U.S. Army has achieved this goal to some degree.

5.4.2 Linking Strategy between CS-LSSAF, CS-VSSAF and CS-CSSAF

The agile roadmap seeks to complete the LVC-ITA within a short time. From recommended action 1 to recommended action 3, we achieved the establishment of defense M&S process, reuse and interoperability of entity (or object) and components, and interoperability of TDB. The last issue for the LVC-ITA is to ensure the interoperability between LVC simulation systems. Accordingly, in Section 4.7.12, we reviewed the state-of-the-art technology that can be connected to CS-SSAFs that we have developed. The recommended action is to develop common bridging capabilities.

In the following section, we present two parallel strategies to ensure the interoperability: (a) short-term strategy and (b) long-term strategy.

5.4.2.1 Strategy 1: Short-Term Strategy

This section covers Short-Term Strategy. In Section 5.2, we developed the CS-SSAF. We researched state-of-the-art technologies that link between CS-LSSAF, CS-VSSAF and CS-CSSAF. The technology is Web-based Technology, which includes the Universal Bridging Tool technology.

5.4.2.1.1 Web-based Technology

This section describes the web-based technology that allows interoperability between M&S simulation environments (DIS exercise, an HLA federation, a TENA execution, or etc.).

With the advances in M&S technology, simulated systems are becoming increasingly sophisticated and increasing complex. Simulation systems can now be accessed via Personal Digital Assistant (PDA) such as smart phones and PC tablets. No longer are simulations constrained to desktop and embedded user interfaces. In Figure 101 below, the simulation systems on the right may be using DIS, HLA 1.3, HLA 1516, HLA Evolved, TENA, or any other protocol for which a Broker exists.

The proposed web-based client technique takes advantage of web service technologies in order to execute complex scenarios within distributed simulation environments. The web-based client technique depends on the network connectivity.

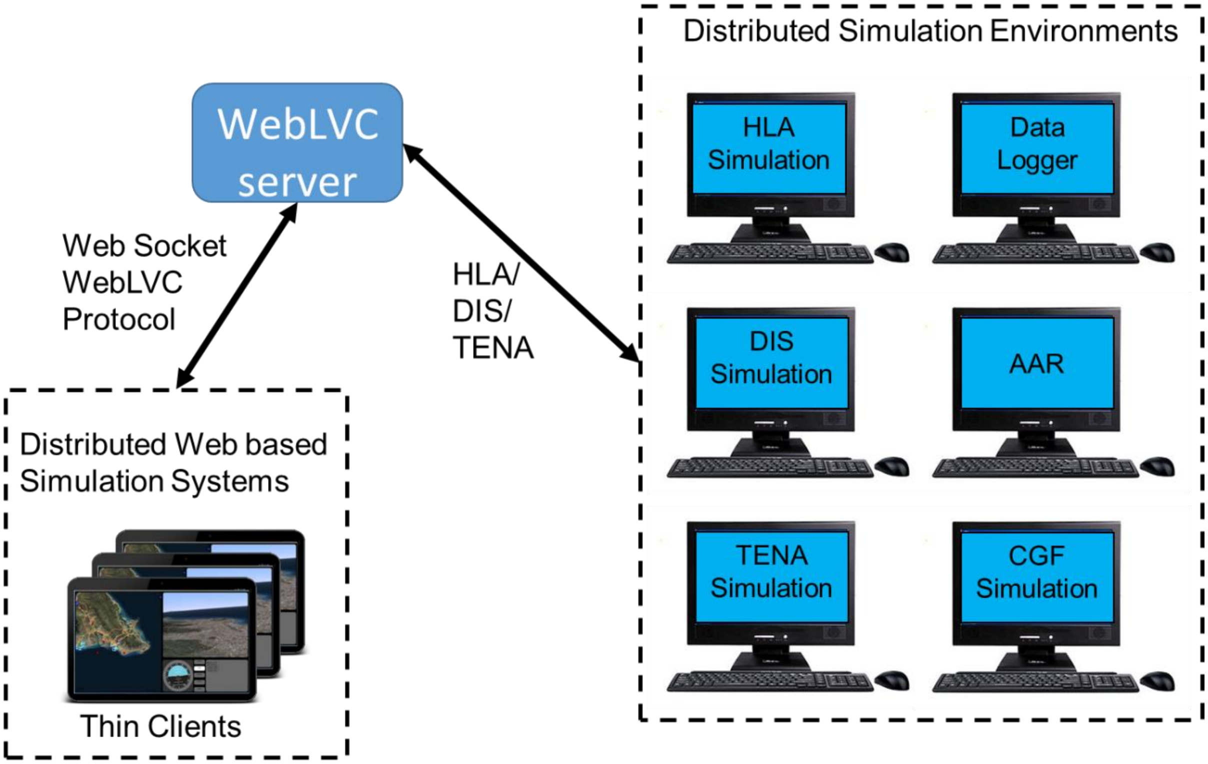


Figure 101: WebLVC Server

In conclusion, when we use the web-based technology above between CS-LSSAF, CS-VSSAF and CS-CSSAF, the overall configuration is shown in Figure 102.

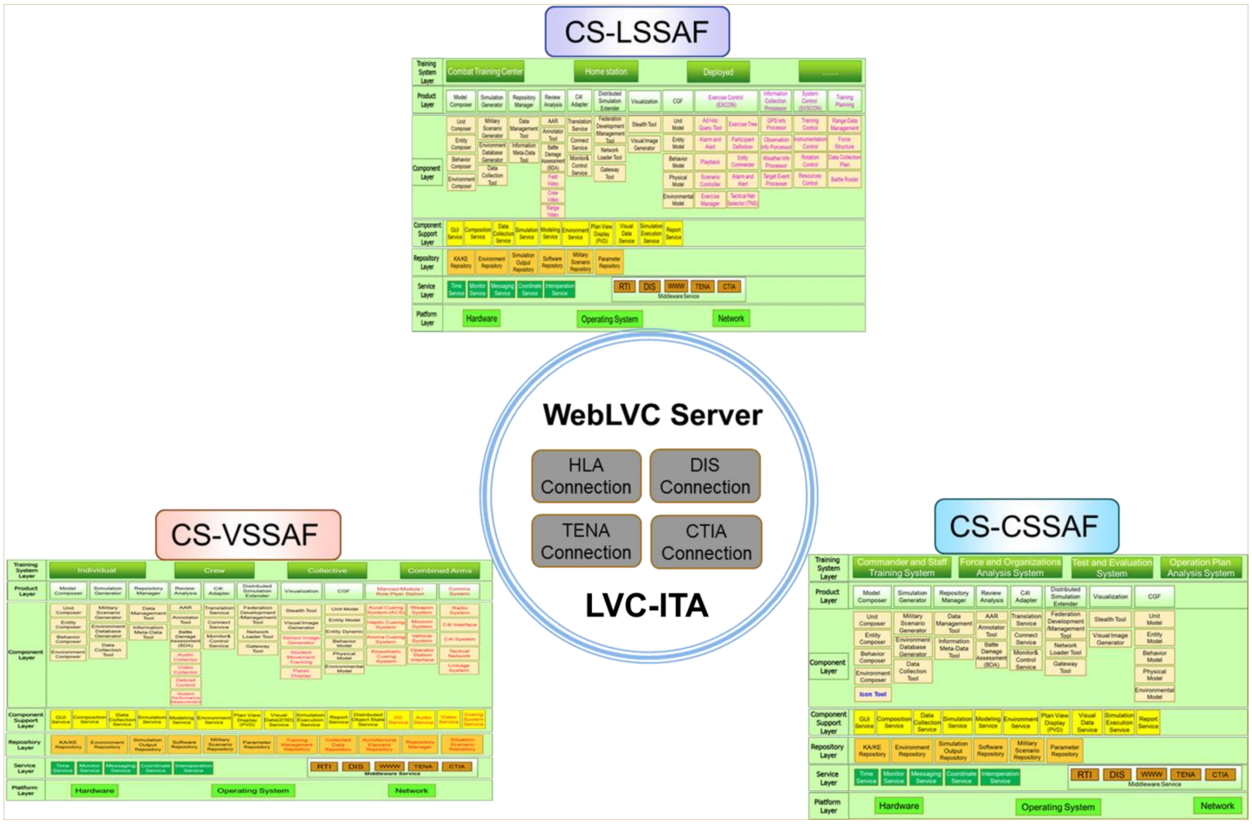


Figure 102: Short Term Strategy for LVC-ITA

5.4.2.2 Strategy 2: Long-Term Strategy

Although we proposed the short-term strategy using the web-based technology including the universal bridging tool. In Section 5.4.2, the final configuration of the LVC-ITA is to pursue a “plug and play” between the CS-LSSAF, CS-VSSAF and CS-CSSAF. Achieving the plug and play method will take a long time because the working is the system of systems process for integrating hardware and software. Figure 103 shows the final ending state of plug and play for LVC-ITA.

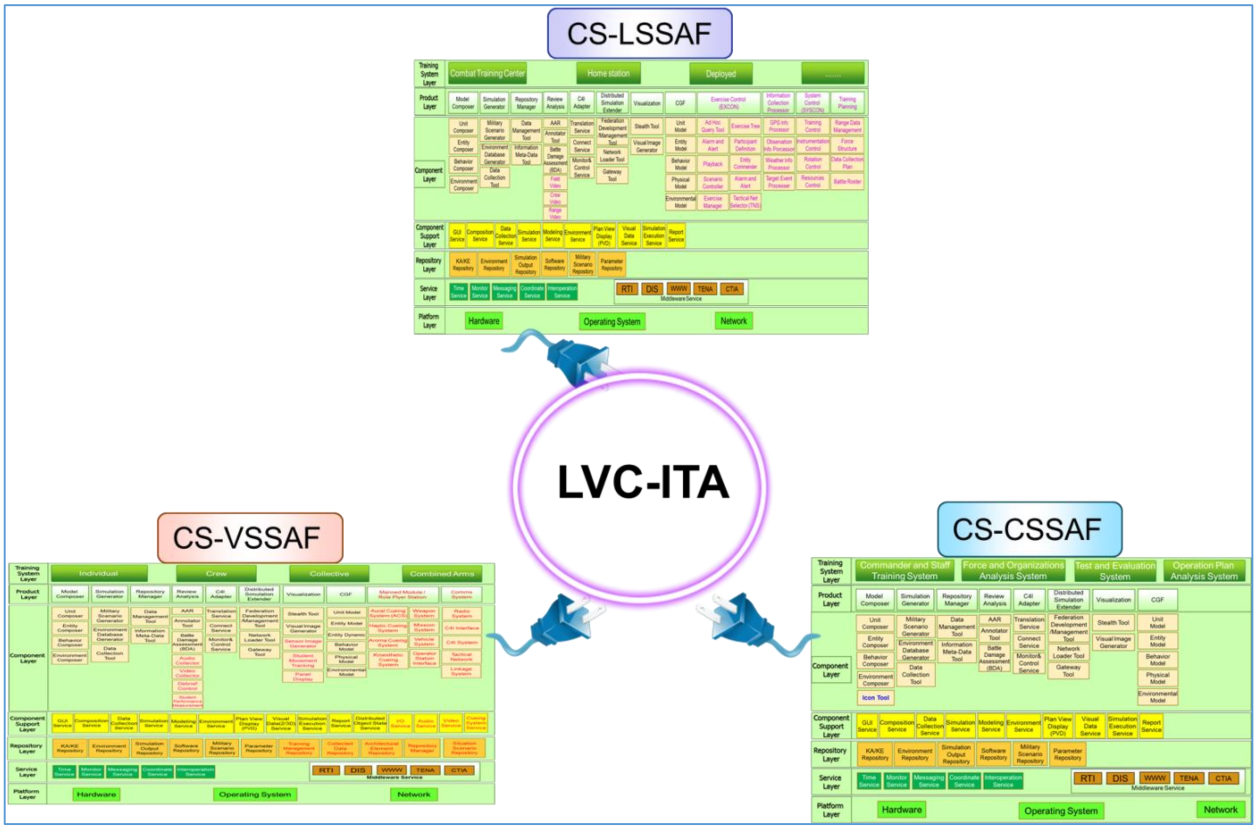


Figure 103: Long Term Strategy for LVC-ITA

CHAPTER SIX: CONCLUSION

This chapter summarizes the results and contributions of my dissertation and suggests future work arising from investigating M&S covering a vast range of advanced technology and concept in this dissertation study.

6.1 Summary

As noted in Chapter 1, the motivating problem of my dissertation is limited LVC simulation. The fundamental reasons of this matter are (a) Inherent Limited Interoperability between the Different Standard Simulation Architectures (SSAs), (b) Many Issues in Integrating LVC Assets, and (c) Decentralized Development of SSAs and LVC Assets. Therefore, we need to research prior or current approaches for seamless LVC simulation.

We reviewed a large amount of relevant literature. In addition, we investigated state-of-the-art technology and skill with respect to interoperability, composability, integration and reuse. In order to answer the research questions, we placed more emphasis on analysis and evaluation of previous methodologies and procedures. Then, we could identify the current state, functional requirement, priority and capabilities for LVC simulations. We identified research gaps as follows: (a) Complex Integration, (b) Long time to LVC user-usage, (c) High cost, and (d) Inflexible integration.

The goal of my dissertation is to provide an *agile roadmap for the Live Virtual Constructive-Integrating Training Architecture (LVC-ITA)*. The methodology for an agile roadmap of the LVC-ITA was composed of four steps in total.

We conducted case study research, because the case study would be the preferred method, when the central research questions are “how” or “why” questions. We wanted to know how to build an agile roadmap for LVC-ITA. Therefore, the objective of the case study was to analyze and evaluate the LVC simulation systems that reflect current M&S technologies. In addition, the case study was to investigate the technologies and methodologies to apply for LVC-ITA from lessons learned. In the case study, an LVC simulation environment was designed to create the *Air- Defense Engagement* scenarios. The case study demonstrated Air-Defense Engagement scenario between Virtual F-16 flight simulator and Virtual SA-8 SAM simulator in SIMbox, Air Defense Radar plyer in AddSIM, VR-Forces for entities representation and Data Logger for AAR. The LVC distributed simulation configuration was based on the DIS and HLA with a target simulation environment. Then, we connected DIS based federation and HLA based federation with WebLVC server.

We evaluated and analyzed the case study’s findings and then identified problems (or limitations) from the case study results. We found eight problems: (a) lack of interactions between simulation entities, (b) lack of reusability, (c) lack of scalability and interoperability of HLA federation, (d) limited capability of the CGFs (or SAFs), (e) limited reference models in database, (f) limited correlated TDBs representation, (g) limited use of the simulation systems for multipurpose, and (h) limited analysis of engagement result.

From the case study results, we learned a great deal about the LVC simulation and drew twelve lessons learned: (a) need for a common standard simulation entity, (b) need for an entity level simulation system, (c) need for common standard defense conceptual modeling, (d) need for CGFs (or SAFs), (e) need for multiple SSAs compliancy, (f) need for scalability capability of simulation systems, (g) need for a common correlated terrain databases (TDBs), (h) need for a new common standard simulation architecture (C-SSA), (i) need for a product line architecture framework (PLAF) concept, (j) need for a simulation system to support multiple M&S applications and LVC simulations, (k) need for a battle damage assessment (BDA) application, and (l) need for a general bridging tool.

To realize these lessons learned, we have developed the following set of four recommended actions: (a) common standard-defense modeling and simulation process (CS-DMSP), (b) common standard-simulation system architecture framework (CS-SSAF), (c) common standard-correlated terrain database (CS-CTDB), and (d) advanced interoperability technology.

6.2 Contribution

The agile roadmap addressed the important issues obtained from the LVC simulation case study. This roadmap provided four recommended actions to be considered as top priority. It is anticipated that this roadmap will eventually lead to an establishment of a full set of common products, data and capabilities that will result in full interoperability, reuse, integrability, composability and seamless set of LVC tools for the M&S developing community (or country).

In the recommended action 1, the agile roadmap first proposed the Common Standard-Defense Modeling and Simulation Process (CS-DMSP) and then discussed the common model.

Through the CS-DMSP, we can enhance the reuse and interoperability. The reuse of simulation object model is a key feature for cost-effective development of simulation environment.

In the recommended action 2, the roadmap also highlighted the Common Standard-Simulation System Architecture Framework (CS-SSAF).

I developed the CS-SSAF as a priority, which is the basis of all LVC architecture frameworks. Based on the CS-SSAF, I developed CS-LSSAF, which is a common standard architecture framework in the Live domain, CS-VSSAF, which is a common standard architecture framework in the Virtual domain, and CS-CSSAF, which is a common standard architecture framework in the Constructive domain. The CS-LSSAF, CS-VSSAF and CS-CSSAF are architectural standard solutions that promote reuse and interoperability for each Live, Virtual and/or Constructive domain.

One of the benefits provided by a CS-SSAF is that systematic reuse, rather than opportunistic reuse, is the major reuse method. By systematically reusing software components, the cost of maintaining and extending components is shared across all of the systems that are using the component. Each of these efforts, are moving forward with systematic reuse initiatives.

Through successful execution of the CS-SSAF strategy, each simulation architecture framework will deliver a set of common components that provide interoperable training solutions for LVC simulation training. The CS-SSAF will facilitate an integrated Live, Virtual and Constructive training environment (LVC-TE).

In addition, through commonality, the CS-SSAF will reduce future development and life-cycle costs. It must support a gradual evolution through a series of incremental path for existing legacy simulation systems.

In the recommended action 3, the roadmap then described the common standard-correlated terrain database (CS-CTDB). I provided the parallel development strategy for developing CS-CTDB. Strategy 1 is to use a legacy simulation system. Strategy 2 is to develop the CS-CTDB with reference to Standard/Rapid Terrain Database Generation Capability (STDGC). Strategy 1 will provide the enhanced terrain correlation and interoperability between the RCS and RVTTS on the same TDB. Strategy 2 provided the guideline for constructing the master TDB generation centers and the CS-CTDB will provide the correlated TDB to not only Virtual domain, but Constructive, Live and Gaming domain.

Finally, in the recommended action 4, the roadmap discussed the policy establishment regarding the multiple SSAs through the comparison between M&S developing communities and the U.S. Army. The WebLVC which is the advanced interoperability technique can ensure the interoperability between the CS-LSSAF, CS-VSSAF and CS-CSSAF.

In conclusion, we can lay the foundation for LVC-ITA through the agile roadmap we proposed as shown in Figure 104.

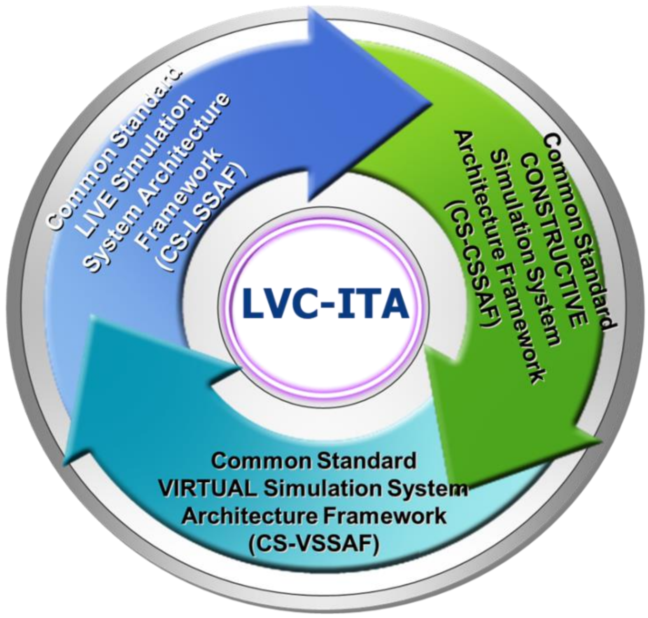


Figure 104: Live Virtual Constructive-Integrating Training Architecture (LVC-ITA)

6.3 Limitations and Future Work

My dissertation study touched on various and important technical issues in LVC simulation to investigate how to make an interoperable, reusable and composable LVC simulation environment. Also, the dissertation study gave us the need for further investigation. Here I specify the list of limitation and future work to be done.

In recommended action 1, the Common Standard-Defense Modeling and Simulation Process (CS-DMSP) need to be extended to represent subtasks of the every step. The complete CS-DMSP will help to construct the defense simulation environment. Next, the Common Model we proposed provided just a conceptual model. I did not demonstrate the successful implementation of these generalizations.

In recommended action 2, the Common Standard-Simulation System Architecture Framework (CS-SSAF) will be developed to support a variety of simulation systems. The CS-SSAF developed through literature review, simulation system SIL hold, and SIL researchers is necessary to be verified from the developers.

In recommended action 3, the Common Standard-Correlated Terrain Database (CS-CTDB) is very important. However, I did not suggest the technical methodologies, although, I provided two parallel strategies for the development of the CS-CTDB system. M&S communities would develop the CS-CTDB systems suited to their environment.

In recommended action 4, I reviewed the advanced interoperability technology. These technologies can be used to interconnect the respective CS-LSSAF, CS-VSSAF and CS-CSSAF mentioned above. For the interoperability between them, I suggested state-of-the-art technologies, but I did not propose the technical methodologies to integrate CS-LSSAF, CS-VSSAF and CS-CSSAF for plug and play. M&S developing communities shall maintain the two strategies that I proposed for parallel development. Afterwards, the integration between them will be achieved.

LIST OF REFERENCES

Abdellaoui, N., Taylor, A., & Parkinson, G. (2009). Comparative Analysis of Computer Generated Forces' Artificial Intelligence: DTIC Document.

Andreas, T., Saikou, D., & Charles, T. (2007). Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. *Journal of Systemics, Cybernetics and Informatics*.

APL. (2010). Guide for Multi-Architecure Live-Virtual-Constructive Environment Engineering and Execution (T. J. H. U. A. P. LABORATORY, Trans.). 11100 Johns Hopkins Road Laurel, MD 20723.

Ball, R. E. (2003). *The fundamentals of aircraft combat survivability analysis and design*: AIAA (American Institute of Aeronautics & Astronautics).

Benali, H., & Saoud, N. B. B. (2010). Towards a component-based framework for interoperability and composability in Modeling and Simulation. *Simulation*.

Bizub, W., Bryan, D., & Harvey, E. (2006). The Joint Live Virtual Constructive Data Translator Framework-Interoperability for a Seamless Joint Training Environment: DTIC Document.

Bizub, W. W., & Cutts, D. E. (2007). *Live Virtual Constructive (LVC) Architecture Interoperability Assessment.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Black, C., Brown, R. F., Levine, S. H., & Sudnikovich, W. P. (2008). Interoperability Problems Caused by Transitioning to a Service Oriented Environment: DTIC Document.

Calvin, J., Dickens, A., Gaines, B., Metzger, P., Miller, D., & Owen, D. (1993). *The SIMNET virtual world architecture.* Paper presented at the Virtual Reality Annual International Symposium, 1993., 1993 IEEE.

Çelik, T., Gökdoğan, G. F., Öztürk, K., & Sarikaya, B. (2012). An HLA-based tactical environment application framework. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 1548512912465993.

CTIA Live Training Product Line (LT2) Overview Briefing. (2006). CTIA Team at PEO STRI. PEO STRI.

Cutts, D., Gustavson, P., & Ashe, J. (2006). *LVC Interoperability via Application of the Base Object Model (BOM).* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Dahmann, J. S., Kuhl, F., & Weatherly, R. (1998). Standards for simulation: as simple as possible but not simpler the high level architecture for simulation. *Simulation, 71*(6), 378-387.

Davis, P. K., & Bigelow, J. H. (1998). Experiments in multiresolution modeling (MRM): DTIC Document.

Degnan, E. (2009). *Update Live-Virtual-Constructive Integrating Architecture (LVC - IA).* Retrieved from http://www.afams.af.mil/shared/media/document/afd-090416-081.pdf.

DoD. (1995). *Modeling and Simulation (M&S) Master Plan*. (DoD 5000.59-P). Department of Defense.

DoD. (2011). *Modeling and Simulation (M&S) Glossary* 1901 N. Beauregard St., Suite 500 Alexandria, VA 22311 Modeling and Simulation Coordination Office Retrieved from http://www.acqnotes.com/Attachments/DoD%20M&S%20Glossary%201%20Oct%2011.pdf.

Dumanoir, P. (2012). *Integrate vs. Interoperate; an Army Training Use Case.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Dumanoir, P., Keller, R., & Koenig, W. (2006). *Network Centric Warfare Requirements-A Live Collective Training Perspective.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Dumanoir, P., Parrish, R., & Sotomayor, H. A. (2007). *LVC Interoperability: Where is the best place to start?* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Dumanoir, P., Pemberton, B. J., & Samper, W. (2004). *OneSAF Interoperability with CTIA-A LVC Connectivity Approach.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Dumanoir, P., & Rivera, J. (2005). *Live Training Transformation (LT2)-A Strategy for Future Army and Joint Live Training.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Faulk, M. A., Fuchs, R. C., Littlejohn, J. T., & Kemper, M. B. A Product Line Approach for the Virtual Domain.

Ford, K. (2005). The Euclid RTP 11.13 Synthetic Environment Development and Exploitation Process (SEDEP). *Virtual Reality*(3).

Fujimoto, R. M. (1999). *Parallel and distributed simulation.* Paper presented at the Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1.

Graniela, B., & Proctor, M. D. (2012). A network-centric terrain database regeneration architecture. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 1548512912444178.

Granowetter, L. (2013). *The WebLVC Protocol: Design and Rationale.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Gustavsson, P. M., Björkman, U., & Wemmergård, J. (2009). *LVC aspects and integration of live simulation.* Paper presented at the 2009 Fall Simulation Interoperability Workshop, Orlando, Florida.

Haight, D. B. (2007). Preparing military leaders for security, stability, transition and reconstruction operations: DTIC Document.

Henninger, A. E., Cutts, D., Loper, M., Lutz, R., Richbourg, R., Saunders, R., & Swenson, S. (2008). Live virtual constructive architecture roadmap (LVCAR) final report. *US DoD, September*.

IEEE. (2003). IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP). *IEEE Std 1516.3-2003*, 0_1-32. doi: 10.1109/IEEESTD.2003.94251

IEEE. (2011). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE Std 1730-2010 (Revision of IEEE Std 1516.3-2003),* 1-79. doi: 10.1109/IEEESTD.2011.5706287

JCATS: Simulation User's Guide. (2003). Lawrence Livermore National Laboratory.

Johnson, M., Ford, R., Shockley, J., Giuli, R., Oberg, S., & Beebe, M. (2004). *Integration of CCTT and JCATS in an LVC exercise.* Paper presented at the Simulation Interoperability Workshop.

Karagöz, N., & Demirörs, O. (2011). Conceptual modeling notations and techniques. *Conceptual modeling for discrete-event simulation*, 179-209.

Kim, D. H., Oh, H. S., & Hwang, S. W. (2013). *Integrating legacy simulation models into component-based weapon system simulation environment.* Paper presented at the Proceedings of the 2013 Summer Computer Simulation Conference.

Lanman, J., Becker, B., & Samper, W. (2009). *Joint service partnership: extending the live training transformation product line.* Paper presented at the Proceedings of the Interservice/Industry Training, Simulation, and Education Conference.

Lee, T., Lee, S., Kim, S., & Baik, J. (2012). a Distributed Parallel simulation environment for Interoperability and Reusability of models in military applications. *Defence Science Journal, 62*(6), 412-419.

Logsdon, J., & Wittman, R. (2007). *Standardization, Transformation, & OneSAF.* Paper presented at the Improving M&S Interoperability, Reuse and Efficiency in Support of Current and Future Forces, Meeting Proceedings.

Loper, M. L., & Cutts, D. (2008). Live Virtual Constructive Architecture Roadmap (LVCAR) Comparative Analysis of Standards Management: Report M&S CO Project.

Loper, M. L., & Cutts, D. (2010). *Comparative Analysis of Standards Management for LVCAR.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

MÄ K. VT MÄ K.   Retrieved 04/06/2015, 2015, from http://www.mak.com/

MÄ K. (2011). *VR-Forces Users Guide*

Marsden, C., Aldinger, M., & Leppard, B. (2009). *Toward Interoperability between Test and Training Enabling Architecture (TENA) and Distributed Interactive Simulation (DIS) Training Architectures.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Mittal, S., Doyle, M. J., & Portrey, A. M. Human in the Loop in System of Systems (SoS) Modeling and Simulation. *Modeling and Simulation Support for System of Systems Engineering Applications*, 415-451.

Mller, B. (2013). THE HLA TUTORIAL v1. 0. *Pitch Technologies, Sweden*.

. *MODELING AND SIMULATION (M&S) MASTER PLAN*. (1995). (DoD 5000.59-P).

Mojtahed, V., Lozano, M. G., Svan, P., Andersson, B., & Kabilan, V. (2005). *DCMF-Defence Conceptual Modelling Framework*: Totalförsvarets forskningsinstitut (FOI).

Morrison, J. E., & Meliza, L. L. (1999). Foundations of the after action review process: DTIC Document.

Morse, K. (2010). Live-Virtual-Constructive Architecture Roadmap Implementation, Common Capabilities-Common Data Storage Formats Implementation Plan: Technical Report for the Joint Training Integration and Evaluation Center. Maryland, USA.

Myjak, M. D., Clark, D., & Lake, T. (1999). *Rti interoperability study group final report.* Paper presented at the Proceedings of the Simulation Interoperability Workshop.

NATO. (2009). *The NATO Modelling and Simulation Standards Profile (NMSSP)*.

Noseworthy, J. R. (2008). *The test and training enabling architecture (TENA) supporting the decentralized development of distributed applications and LVC simulations.* Paper presented at the Distributed Simulation and Real-Time Applications, 2008. DS-RT 2008. 12th IEEE/ACM International Symposium on.

O'Connor, M., DiCola, J., Sorroche, J., Lane, J., Lewis, D., & Norman, R. (2006). *A Mixedarchitecture for Joint Testing.* Paper presented at the the Proceedings of the 2006 Spring Simulation Interoperability Workshop. Huntsville, AL.

Page, E. H., Briggs, R., & Tufarolo, J. A. (2004). *Toward a family of maturity models for the simulation interconnection problem.* Paper presented at the Proceedings of the Spring Simulation Interoperability Workshop.

PEO-STRI. (2006a). *Common Standards, Products, Architectures and/or Repositories (CSPAR) Baseline Document* Orlando: U.S. Army Program Executive Office (PEO) Simulation, Training and Instrumentation (STRI).

PEO-STRI. (2006b). *Synthetic Environment (SE) Core*. Retrieved from http://www.peostri.army.mil/.

PEO-STRI. (2012). SE Core. from http://www.peostri.army.mil/PRODUCTS/SECORE/

PEO-STRI. (2013). Close Combat Tactical Trainer (CCTT). from http://www.peostri.army.mil/PRODUCTS/CCTT/

Petty, M. D., & Weisel, E. W. (2003). *A composability lexicon.* Paper presented at the Proceedings of the Spring 2003 Simulation Interoperability Workshop.

Pidd, M. (2003). Tools for Thinking: Modelling in Management Science. 2003. *John W iley & Sons, Chichester*.

Rieger, L. A., & Lewis, J. (2006). *Integrated Middleware for Flexible DIS and HLA Interoperability.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Rivera, J., Samper, W., & Clinger, B. (2007). *Applying the Live Training Transformation (LT2) Software Reuse Strategy to the Homestation Instrumentation Training System.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Rivera, J., Samper, W., & Clinger, B. (2008). *Live Training Transformation (LT2) product line applied standards for reusable integrated and interoperable solutions.* Paper presented at the Military Communications Conference, 2008. MILCOM 2008. IEEE.

Robinson, S. (2008). Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society, 59*(3), 278-290.

Rumpel, B., & Vila, R. (2007). *Distributed Training in Europe.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Shanley, M. G. (2007). *Supporting Training Strategies for Brigade Combat Teams Using Future Combat Systems (FCS) Technologies* (Vol. 538): Rand Corporation.

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.* Paper presented at the 98 Spring Simulation Interoperability Workshop Papers.

Shufelt Jr, J. W. (2006). A vision for future virtual training: DTIC Document.

SimiGon. SIMbox Version 5.6 Release Notes [Press release]. Retrieved from http://www.simigon.com/pdf/releasenotes_56.pdf

SimiGon. SimiGon. from http://www.simigon.com/

SISO. (2007). Reference for Guide: DIS Plain and Simple.

Steinman, J., & Parks, J. (2007). *A Proposed Open System Architecture for Modeling and Simulation (OSAMS).* Paper presented at the SISO Simulation Interoperability Workshop. Orlando, FL.

Steinman, J. S. (2013). The Roadmap. *Simulation Interoperability Workshop*.

Steinman, J. S., & Hardy, D. R. (2004). Evolution of the standard simulation architecture: DTIC Document.

Tolk, A. (2012). *Engineering Principles of Combat Modeling and Distributed Simulation*: John Wiley & Sons.

Tolk, A., & Muguira, J. A. (2003). *The levels of conceptual interoperability model.* Paper presented at the Proceedings of the 2003 Fall Simulation Interoperability Workshop.

U.S._Naval_Research_Laboratory. SIMDIS. from https://simdis.nrl.navy.mil

Wang, W., Tolk, A., & Wang, W. (2009). *The levels of conceptual interoperability model: applying systems engineering principles to M&S.* Paper presented at the Proceedings of the 2009 Spring Simulation Multiconference.

Weatherly, R. M., Wilson, A. L., Canova, B. S., Page, E. H., Zabek, A. A., & Fischer, M. C. (1996). *Advanced distributed simulation through the aggregate level simulation protocol.* Paper presented at the System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on.

Wittman Jr, R. L., & Harrison, C. T. (2001). OneSAF: a product line approach to simulation development: DTIC Document.

Yin, R. K. (2014). *Case study research: Design and methods*: Sage publications.

Zalcman, L., Blacklock, J., Foster, K., & Lawrie, G. (2011). An Air Operations Division Live, Virtual, and Constructive (LVC) Corporate Interoperability Standards Development Strategy: DTIC Document.

Zeigler, B., Praehofer, H., & Kim, T. (2000). Theory of Modeling and Simulation, 2000: Academic Press.