BOOKS ON COMBAT MODELING

Rs. 10 Army Combat Modeling Air Force Combat Modeling 10 10 Naval Force Combat Modeling Amphibious Force Combat Modeling 10 Air-Land Combat Modeling 10









Army, Navy, Air Force

Ministry of Defense Delhi, India



www.mbsc.org

Strength is Life

SUMANTA KUMAR DAS ⊛(8) MODEL BASED SYSTEM DESIGN

SUKUD

Using System Composer Toolbox in MATLAB



Complex System Engineering Operations



DESIGN

SUMANTA KUMAR DAS

Toolbox in MATLAB MODEL BASED SYSTEM DESIGN

Using System Composer

Complex System Engineering Operations



Published by

Ministry of Defence, Delhi, India from its Publication House, Delhi Email: @gmail.com
Website:www.mbsc.org

© Ministry of Defense
All rights are reserved and exclusively
Vested with the right-holder. No part of this work may
be reproduced, stored, published, circulated,
distributed, communicated, adopted, and translated in
any material form (by electronic or mechanical
means) without the prior written permission of the
right-holder. Nothing herein prevents a person from
making such uses that are permissible under law.

Seventh – Reprint, November 2023 10M

> Printed in India at Udbodhon Karjalay Delhi-110023

CONTENTS

		PAGE
1	Scope	4
2	Introduction	6
3	Review of Literatures	22
	Reference Map	
4	System Composition	43
	Design Constraints, Design	
	Methodology	
	System Engineering and Requirement	52
	Identification	
	Architecture, Components, Properties	56
	and Interfaces of the System	
	Overall System's Architecture and	60
	Linkages	
	Multiple System View	93
	Categorization of the System	105
	Conclusion	113
5	Appendix I: Force Ratio	109
6	Appendix II: Lanchester's Models	111
7	Abbreviations	115
	Glossary	117
	Index	131

PREFACE

This book is written specifically for f i v e different p u r p o s e s which expands its scope starting from d e s i g n i n g to v e r i f i c a t i o n a n d v a l i d a t i o n of a new system. The KNOWLEDGE shared in this book can be utilized

- to design and develop a common framework of modelling and simulation of STRATEGIC and OPERATIONAL level collaborative operations involving at least MORE THAN ONE domain of Services starting from land, water, air, space, information, and cyberspace and considering different combat resolution scenarios as case study.
- to estimate the parameters of a g g r e g a t e d force-on-force combat attrition model from the high-resolution simulation data of different military operations pertaining to air, water, ground and amphibious domain.

- to integrate the effect of space, information and cyber components as force multiplier over the aggregated force-on-force combat model
- to estimate the relative f i r e p o w e r s c o r e values of different components in a military operation using their system characteristics.
- to verify and validate (V&V) the estimated models with h i s t o r i c a l b a t t l e s wherever data is available from open literatures, otherwise following the simulation based approaches, multi-criteria decision making or expert judgement technique for V&V.

In addition to that this book can be utilized to address the issues related to transforming the estimated models of different military operation scenarios into a deliverable gaming system.

Happy reading!

SuKuDa

New Delhi Date:



CHAPTER 1

1. Introduction

Model Based System Composition (MBSC) is the emerging technique of system development that assist in developing system's architectures based on the modeling requirements. This architecture is used for designing, analyzing, and testing

the system. Defense planners and decision makers use warfare models and simulation ¹ to predict likely outcomes of combat. These warfare models are generally represented in the form of mathematical equations (e.g., a system of deterministic differential equations or adaptive difference equations), which represent the gradual interaction and attrition processes of the two sides.

Simulation is commonly used to study large, complex systems like combat. There are two different representations for simulating a system through time: the "fixed time-step" and the "event-to-event". Most of the simulations of combat force use both the time-step method for the target-acquisition process and the event-step method for all other processes. The basic difference between time-step approach and event-step approach is about the form of time frame. The time-step approach has emphasized the use of continuous differential time in the models whereas event based simulation

¹ is an abstract representation of a physical system, situation, or phenomenon (Jaiswal, 1997).

emphasized the conflict as series of discrete events (discretetime process).

In modern times the term collaborative operation is very common in defense planning process. By Collaborative operation we mean that any military operations that are being conducted in coordination with air force, navy, army, space and cyber command. Although in our country there is no dedicated command for space and cyber, in future we may have separate command for space and cyber. The basic structure of any Collaborative operations involve an O O D A (Observe-Orient-Decide-Act) loop or variants (F. Bolderheij, 2005), (Stromberg & et., 2002). Although, the OODA loop was initially originated from behavioral science, latter it was exploited for understanding the human participations in complex military problems. Along with the OODA loop recently, the object oriented approaches are also becoming popular because of its enhanced capability of practical reasoning for developing modern software systems. It has the advantages from the user

perspective in terms of both speed and ease of development of models.

In recent times information sharing and collaborative decision making over the defense networks have completely revolutionized the combat scenarios (D. S. Alberts, 1999). Today's offensive forces are equipped with sophisticated electronic attacking (EA) or electronic counter measuring (ECM) devices (for electronic jamming against radar and communications), Early Warning and Controlling System (EWACS), high speed of information flow, high precision air-to-air, air-to-surface missiles, high speed fighters, bombers, unmanned air vehicles (UAV) etc. To respond to these, the defensive forces rely on early warning surveillance, tracking, over-the-horizon (OTH) radar, orbital military satellites, counter surveillance and reconnaissance system (CSRS), counter communication system (CCS), interceptor missiles, cyber intelligence, that has electronic iamming counter counter measure (ECCM) anti technologies, high-tech command and controls (C2) that robustly assess the situation and efficiently decide the best course of action (COA). Modeling such combat processes is of utmost importance to survive with such technological and collaborative advancement.

The collaborative Army, Air Force & Naval Game is an aggregation of sensors, weapons, C2, intelligence systems, communications, and personnel operating under the Collaborative Task Force (CTF) of a designated Joint Commander. The gaming systems have progressed steadily over the recent years to include highly sophisticated computer-based software systems to assist and train the Commander. Some of the examples of such tools are Air Force Mission Support System, PowerScene, TopScene etc. (D. S. Alberts, 1999).

An effective plan to exercise a collaborative war game involving the army, navy and air force, the primary unit that needs to be modeled in the game may be a Brigade or Division for the Army, an Air Squadron for the Air Force and a Task Group for the Navy. Each of these larger units comprises of heterogeneous forces concerning everything starting from the weapon system to logistic resources as well

as command and control. It has been seen that most of the aggregated models developed so far are for the theater (Hillestal & Moore, 1996) or campaign (Hillestad, Bennett, & Moore, 1996) level game and most of the game is played as human-computer interactive simulation. Generally, Multiple decision-makers are involved in those game.

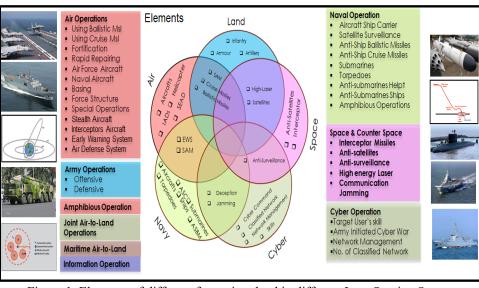


Figure 1. Elements of different forces involved in different Inter Service Ops.

Most of the wars that have been taken place in the world in the recent times have seen the Air Force, Navy and Army fight together. Most of the countries in the world are forming a collaborative force consisting of the Army, Navy, and Air Force. Countries are preparing themselves in a way that the enemy can be defeated in water, land, and air. From the analysis of data from open literatures, today we can predict the outcome of the war before the war really begins. That requires a thorough analysis of the influencing factors or elements or equipment of strategic and operational scenarios.

Table 1. Factor charts showing involvement of different factors in Inter Service operations

CL N-	Operational Scenarios	Factors**										
SI. NO.		1	2	3	4	5	6	7	8	9	10	
1	Air Base Attack using Missiles	X1	X2	Х3	X4	X5	X6	X7	X8	X9	X10	
2	Airbase Attack by Aircrafts	X11	X12									
3	Air Superiority	X13	X14	X15	X16	X17	X18	X19				
4	Airspace Penetration	X20	X21	X22	X23	X24	X25	X26	X27			
5	Defender Anti-surface Warfare	X28	X29	X30	X31	X32						
6	Attacker Anti-Surface Warfare	X33	X34	X35	X36	X37						
7	Attacker Counter space	X38	X39	X40	X41	X42						

Sl. No.	Operational Scenarios	Factors**									
		1	2	3	4	5	6	7	8	9	10
8	Defender Counter space	X43	X44	X45	X46	X47					
9	Cyber War	X48	X49	X50	X51						

** X1: No. of Cruise Missiles, X2: No. of Ballistic Missiles, X3: Range of Cruise Missiles, X4: Range of Ballistic Missiles, X5: CEP of Cruise Missiles, X6: CEP of ballistic Missiles, X7: Fortification Factor of Defender, X8: runway repair method, X9: Aircraft Dispersion, X10: Basing, X11: Varieties of PGMs, X12: Number of Stand-off Weapons, X13:Generation of Air force Aircraft, X14: Generation of Navy Aircraft, X15: Basing, X16: Flight Distance, X17: Force Structure, X18: Period of the Campaign, X19: X20: SAM, X21:Seeker, X22: Ranger, X23: Sophisticated aircraft, X24: Airborne early warning aircrafts, X25: IADs, X26: Stealth Aircraft, X27: SEAD, X28: Satellite link to monitor aircraft carrier, X29: Anti-ship ballistic missiles, X30: Submarines, X31: Torpedoes, X32: Cruise missiles, X33: Amphibious Ships, X34: anti-submarine Helicopter, X35: Anti-submarine ships, X36: air force cruise missiles, X37: naval cruise missiles, X38: Number of Orbital military satellites, X39: Counter Surveillance Reconnaissance System (CSRS), X40: Tactical High energy Laser system, X41: Interceptors, X42: Counter Communication System (CSRS) of defender, X45: Tactical High energy Laser system of defender, X46: Interceptors of defender, X47: Counter Communication System

CL N	Operational Scenarios	Factors**									
SI. No.		1	2	3	4	5	6	7	8	9	10
(CCS) of defender, X48: No. of Cyber Command, X49: Cyber skills of the side, X50: network management, X51:											

(CCS) of defender, X48: No. of Cyber Command, X49: Cyber skills of the side, X50: network management, X51 Number of classified Network.

In different strategic operations, it is very important to make a scorecard of how different equipment will determine the outcome of a war. In this section we have a general factor-chart, considering prepared collaborative operations scenarios that have been thoroughly analyzed and important factors have been identified. To prepare this factor-chart different combat attrition models are considered. This factor-chart is made up of the factors that the commander has to keep in mind while conducting various collaborative inter Service operations. Just as it is difficult to calculate how much damage can be done in any Collaborative operation, it is much easier to compare the relative strength of two combating forces. The following table enlist ten types of collaborative operation scenarios along with the name of the mathematical models which will be required to simulate these collaborative Operations.

Table 2. Models required to simulate different collaborative co-operational scenarios

Sl. No.	Generic Operational Scenarios	Models Required						
1	Air Base Attack by SRBM ¹	Ballistic Missiles attack on air bases						
2	Airbase Attack by Bombs/PGM	Runway Attack Model						
3	Air Superiority	Air Combat Model						
4	Air space Penetration	Air space penetration Model						
5	Defender Anti-surface Warfare	Effectiveness of Submarine						
6	Attacker Anti-Surface Warfare	Effectiveness of Submarine						
7	Attacker Counter space	Effectiveness of missiles Interceptor						
8	Defender Counter space	Effectiveness of missiles Interceptor						
9	Cyber War	Network Management Model						
SRBM: S	SRBM: Short Range Ballistic Missile							

Figure 2-5 illustrates different conceptual structure of the system. The system consists of ten major simulation models and each of these consists of mathematical models, data and operational philosophy translated into algorithms and decision rules. These subsystems are:

Code No.		Combat Models	What & Why					
M1	Land W	arfare Models	Ground Combat					
	M1.1	QJMA	Force Quantification [(Dupuy (1985)]					
	M1.2 LANCHESTER M1.3 ADM		Dynamic Aggregated Combat Model					
			Dynamic Attrition Rates					
	M1.4	SFS	More Detailed Aggregated combat Model					
M2	Anti Surface Warfare Models		Naval Ship to ship battle					
M3	Airbase Attack using Missiles		Airbases near conflict area is most critical point					
M4	Airbase	Attack using Air Power	Capability to Neutralize Enemy Offensive Air Power					
M5	Air Sup	eriority	Which side is air superior					
M6	Air Pen	etration	Which side has air penetration capability					
M7	Air Défe	ense Models	To model threat assessment and weapon allocation					
M8	Amphib	ious Models	Joint Amphibious operation execution					
M9	Cyber		Role of Cyber power in Attack and Defense					
M10	Space		Role of Information for execution of Joint Operation					

Figure 2. Modeling requirements of the system

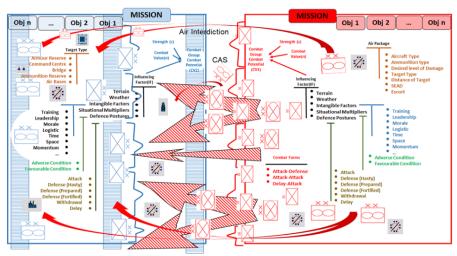


Figure 3. An abstract view of the System

System Design

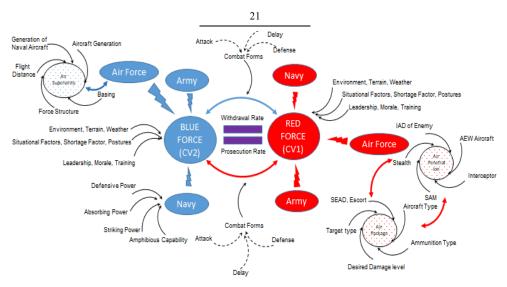


Figure 4. Influencing factor of different modules of the system

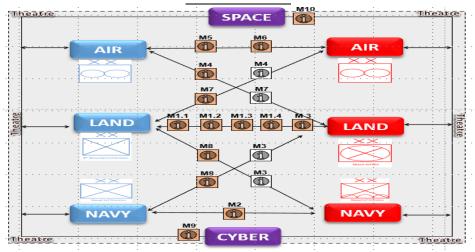


Figure 5.Conceptual representation of the system environment (also refer figure 2 for details about the models)

CHAPTER 2

2. Previous Trends Affecting the Current System

In July 12, 2018 a report "Joint doctrine" (IDS, 2018) was unveiled in India that provides the detailed Operational Synchronization among the Army, Navy and Air force with an aim to deal with all possible security threat in India. The document listed a list of all possible scenarios starting from asynchronous war in J&K to mountainous or guerilla warfare in various part of the country.

In that document it was mentioned that for an intertwined collaborative operation, the commanders are required to be familiar with wide spectrum and nature of wars in the future. At every stage of the operation, they must face with new challenges. Therefore, before going for actual battle, training is very important for them to cope

with such difficult situations. To help them think through their options when faced with a force employment decision while applying their knowledge, experience and judgment, military staffs use a methodology called the Joint Operation Planning Process (JOPP) (Joint Operation Planning Process (JOPP) Workbook, July, 2013).,(ISO/IEC/IEEE:15288, 2015)

JOPP can be supported by simulation models of large-scale combat contributing useful insights for many military decision-making problems. The designers of such models attempted to achieve a representation of warfare that is as accurate and believable as possible. For moderate sized forces believability is aided by modeling in high resolution, but larger forces require aggregation to keep the models within the limits of computer size and execution time (Caldwell, Hartman, Parry, & Washburn, 2000), (IEEE:1220-2005, 2005)

planners and decision makers requires Defense mathematical models to predict likely outcomes of combat dynamics. These mathematical models are generally represented in the form of a system of deterministic differential equations, which represent the gradual interaction and attrition process of the two sides. Lanchester in 1914 first introduced the concept of combat modeling using differential equations (Lanchester, 1914). (IEEE:15288.2-2014, 2014) Many analysts have subsequently modified his original work to represent combat dynamics in modern warfare. Weiss (H.K. Weiss, 1975), (ISO/IEC/IEEE:15289:2015, 2015) modified Lanchester's original work for aimed fire (such as by armour). (H. Brackney, 1959), (ISO/IEC-TR24748-1:2010, 2010) introduced the concept of area fire (such as by artillery). (R.L. Helmbold, 1965), (ISO/IEC-TR24748-2:2011, 2011) has given a general form for homogeneousforce attrition rates (square / linear / logarithmic) and proposed a modification of Lanchester equations for

Previous Trends Affecting the Current System

modern warfare to account for inefficiencies of scale for the larger force when force sizes are grossly unequal.

Obtaining numerical values of attrition-rate coefficients (the rate at which an individual weapon-system type kills enemy targets of a particular type) is a major problem for applying the Lanchester model in practice. Two approaches have been originated in this respect (J.G. Taylor, 1983), (ISO/IEC-TR24748-3:2011, 2011):

- use of analytical sub models, of the attrition process to compute the desired numerical values; and
- a statistical estimate, based on 'combat' data generated by a detailed combat simulation.

In reality, actual historical combat data is not easily available. Therefore, the practice is to use data generated either by combat field experiments or by a high-resolution combat simulation. In the latter approach, one uses combat

data to compute statistical estimates of the attrition rate coefficients.

There are four principal statistical methods for computing such point estimates (V.K. Rohatgi, 1984), (ISO-10303-AP233): (a) method of moments estimation (MME) (b) maximum likelihood estimation (MLE) (c) Bayes estimation (BE) and (d) least square estimation (LSE). Of these four methods, only maximum likelihood estimation method has been used extensively for estimating attrition rate coefficients from combat simulation [(G.M. Clark, 1969), (J. Andrighetti, September, 1973.), (R. G. Stockton, 1973), (B.C. Graham, June 1979), (J.G. Taylor U. Y., 2000), (IEEE:1220-2005, 2005)]. Since the original work of (G.M. Clark, 1969), no significant theoretical improvement in the combat simulation approach has appeared in the open literatures. Clark in his work had assumed that every target type on a side had the same target availability for estimation of model parameters. He used the time gap between two successive casualties in his simulation for statistical

estimation of the model parameters. There are no alternatives to such assumptions. However, Taylor (J.G. Taylor U. Y., 2000) has shown how to estimate attrition rate coefficients, without if all target types on a side have the same target availability.

Comparison of high and low-resolution models and the need of aggregation are elaborated in the literature (Caldwell, Hartman, Parry, & Washburn, 2000), (ISO/IEC/IEEE:15288, 2015). High-resolution modelling involves detailed design, high-resolution knowledge usage, narrow in-depth analysis for accuracy, reasoning and comprehension at a more atomic level, and simulating reality. Contrary to this, low-resolution modelling involves simplistic design, low-resolution knowledge responding to mainly high-level questions, reasoning and comprehension with high-level variables, and abstracting "big picture". High-resolution modelling can also be used for informing, calibrating, or explaining low-resolution work

The taxonomy of models, high-resolution and lowresolution models, is widely discussed in the literature (Allen, 1997), (Naval System Engineering Guide, 2004). The focus of these works is on the connection between the strategic planning with detailed analysis. Aggregation and disaggregation are techniques that facilitate interactions at the same level of interactions. The literatures (H.K. Weiss, 1975), (INCOSE, 2020) illustrates common approach used in aggregation using Lanchester theory as a basis. Requirements for theoretically consistent aggregation, disaggregation, and partial aggregation have also been described. Consistency of aggregation and disaggregation in models of combat is a very desirable property. The map required for running a large-scale Collaborative operation through aggregated combat model is generally of coarserresolution. Map characteristics that are important for aggregated modeling are terrain height, movability by the forces, whether the presence of bridge or rail lines, etc. The map preparation for conducting aggregated simulations to accommodate a bigger combating unit is another research topic. Generally, maps of aggregated combat simulations are divided into smaller segments. If we consider the ATCAL² models of JTLS³ and (NASA-SP-2016-6105, 2016), we can see that the operations within one category are completely independent of the other categories. The following table illustrates some of the special characteristics of aggregated combat modeling approaches that have been utilized in various Wargame.

Table 3. Characteristics of few popular aggregated models for collaborative Inter Service model-based system

Name	Terrain Model Units	Movement Model Factors	Speed depends
ATCAL	Segments	Destination	On Unit Type
IDAGM	Sector	Route	Traficability Obstacles or
VECTOR 2	Segments	Speed	Minefield In combat or
JTLS	Segments	Route	not Postures
FORCE	Segments	Speed	Force Ratio

² Aggregated Theater Level Calibration Model

³ Joint Theater Lanchester Simulation

Environmental modeling is another aspect of aggregated combat modeling. The environmental factors that have the most prominence is the visibility, cloud cover, obstacles, day or night conditions, etc. Some collaborative wargame simulation model like COSAGE model has the scope to give the input for two different time scale of day or night. With changing day and night conditions, the target detection probability, target location, lethality of weapons, effectiveness of smoke and illumination rounds also changes. We also need to think about how the night vision devices and other electro-optical devices are enhancing vision capability and how their effects can be reflected in the aggregated modeling. An example of such modeling is VECTOR 2 (Caldwell et. al. 2000) model. It considers environmental conditions for a different combination of bigger fighting units with changing visibility categories with different target and observer combinations. These influences mobility and target acquisition. COMMANDER model (Caldwell et. al. 2000) (for Air

operation) uses 250 nm of ground cover for monitoring weather conditions.

Why do we need aggregated models? High-resolution simulation models are commonly used to represent realworld scenarios. High-level simulation or detailed simulation models cannot be used if we want a simulation that can handle higher force levels. Then it became imperative to model more and more of the bigger forces, in that case, the Aggregated Model is required. The first step in aggregated modeling is to identify what the minimum resolution should be in the model, brigade, division, and so on. After identifying it, it has to be seen what are the smaller sub-unit that constitutes the bigger unit. The next step is to extract an average of the stochastic processes that are being used in the high-resolution simulation and apply it to the aggregated level for the interactions of larger units.

This is why most of the time we see that the aggregated models are being designed at deterministic model. Another important aspect of the aggregated model is its Event

management with the time frame. The time resolution of aggregated models is usually varying from one minute to one day. In that case, not all Event details are available. An average idea is given that there will be some Events during this time frame. Open literatures refer several large-scale aggregated models over the years. The table below gives a comparison study of all simulated aggregated models.

Table 4. Comparison study of different Aggregated Models

Models	Nature	Scale	Year	Domain	Levels
ATLAS	Deterministic	Theater	1960	Army/Air	Div/Mission
CEM	Deterministic	Theater	1970	Army/Air	Brig/Div
IDAGAM	Deterministic	Theater	1974	Army/Air	Brig/Div
VECTOR 2	Deterministic	Theater	1976	Army/Air	Battelion
FORCEM	Deterministic	Theater	1980	Army/Air	Div
COMMANDER	Deterministic	Theater	1980	Army/Air	User Choice
JTLS	Stochastic	Theater	1984	Army/Air	Div/brig
ICOR	Deterministic	Core	1970	Army/Air	Bat/Coy
COSAGE	Stochastic	Theater	1980	Army/Air	Div
FORCE	Deterministic	Division	1970	Army/Air	Bat/Div
				1	

Like aggregated terrain and movement model, combat attrition models are an important issue for conducting collaborative wargame simulation. Two approaches of attrition model are generally adopted. One is the Force Ratio (Allen 1997, Ramazen 2000) and the other is the Lancaster based model (Lanchester 1914, Helmbold 1965, Weiss 1975, Taylor 1983, Turkes 2000). Recently the Salvo model (Hughes, 1993) are also become popular for naval warfare modeling.

Although simulation is a very powerful and useful tool for solving some problems, it is not always the most appropriate method for others. Elaborate combat simulations, which rely upon high fidelity modeling, may not be responsive to rapidly changing technological advances in weapon systems and warfare concepts. The collaborative war gaming system requires a simple model that is capable of capturing the innate properties of different weapon systems and allowing decision makers to understand the relationship and interaction between forces. If we can capture and model the essential characteristics of

Previous Trends Affecting the Current System

these weapon systems, we may be capable of providing easy, but limited, analysis. Alternative to complex simulation for exploratory analysis the Lanchester model and salvo model are such alternatives. It gives the analyst a simple method in which to evaluate the effects of strategic or operational level analysis between two opposing collaborative forces and allows simple insight into the broad characteristics of the battle. These types of models calculate the fraction of combat group, or a collaborative force of identical or heterogeneous components, put out of action as a result of successful engagement from an opposing combat group or collaborative force.

Besides the Lanchester's equations, another approach for combat attrition is models that use force ratio in their structure. The study by (US Army's Center for Strategy and Force Evaluation, September 1998) focuses on aggregate attrition methodologies that use combat power ratio to compute the casualties of the forces. Unlike the Lanchester's equations, there is no study in the literature

that used firepower score attrition models on real data in which force sizes are available day by day for both sides.

The method of determining the firepower scores is a very difficult problem. There are several methods of computing firepower score values, such as military judgement and experience (RAND's ground force scoring system (Allen, 1997)), historical combat performance derived from WWII and the Korean War, and results from high resolution simulations (e.g., Anti-Potential-Potential Method) [(Parry, 1992)]. (Ramazan, 2000) applies several Firepower Score attrition algorithms to the battle of Kursk, WW II data. These algorithms are used in highly aggregated combat models to predict attrition and movement rates. The quality of the available historical data for validation of attrition models is poor. Most accessible battle data contain only starting sizes and casualties, sometimes only for one side. A detailed database of the Battle of Kursk of World War II, the largest tank battle in history, has recently been developed by Dupuy Institute (TDI). The data is two-sided, time phased (daily), highly

detailed, and covers 15 days of the campaign. According to combat engagement intensity, three different data sets are extracted from the Battle of Kursk data. RAND's Situational Force Scoring, Dupuy's QJM and the ATLAS ground attrition algorithms are applied to these data sets. Fitted versus actual personnel and weapon losses are analyzed for the different approaches and data sets. None of the models fits better in all cases. In all of the models and for both sides, the Fighting Combat Unit Data set gives the best fit. All the models tend to overestimates battle casualties, particularly for the Germans.

In addition to the Lanchester and Salvo equations, another approach for combat attrition is force ratio-based attrition model. This method focuses on aggregate attrition methodologies that use combat power ratio to compute the casualties of the forces. Similar to Lanchester and Salvo equations, there is various studies in the literature that used firepower score attrition models on real data in which force sizes are available day by day for both sides.

The system is aimed to simulate different collaborative operation scenarios namely air operation, naval operation, army operation, amphibious operation, collaborative airto-land operation (JALO), marine-air-to-land operation (MALO), cyber warfare, information warfare, special operation warfare. The air-based operation will be mainly conducted by the air power of the side. The elements in airpower are fighter aircraft, strike aircraft, bomber, transport aircraft, helicopter, attack helicopters, air-borne early warning aircraft etc. The equipment's in naval forces are mainly aircraft carrier, destroyer ships, frigates, corvettes, submarines, coastal petrol craft, mine warfare crafts. The Army dominated collaborative operation will be mainly conducted by the ground forces (Infantry, armour and artillery) of the side. Similarly, amphibious operation will be conducted by the collaborative amphibious forces comprising of landing vehicle assault craft (LVAC), its associated support force, infantry personnel, Tank, ATGM etc. The collaborative air-to-land operation (JALO) is done in coalition of ground-based army and air power of the collaborative task force. The elements of this operation are Missiles (ballistic, cruise), aircrafts, bombers and other related air elements. Similarly, the maritime-air-to-land operation (MALO) is done in coalition of marine and air power. Therefore, aircraft carrier, ships, destroyer, IADs will be integral part of this operation. In cyber operation we will be considering spread of malware as the self-replicating agents which will directly or indirectly influence the force effectiveness. In case of space-based operation, the role of satellite for intelligence and surveillance will be considered.

The modeling approaches will be aggregated combat model based on Lanchester, salvo and force ratio-based approaches. Entire Modelling tasks of the system proposed has been categorized into 9 broad categories based on the major domain of operations. The main part of the *system* modelling task is the attrition model and its associated data. The attrition model which we will be considering will be at the aggregated level specifically Lanchester's based differential equations. To modify the basic Lanchester's

equations to fit in a particular operational scenario we have to study the other related influencing factors which will be affecting the basic models. In addition to the basic domain operations (army, navy, air) other components of the collaborative operations like cyber, information and space components are also being tried to include in the model. Most of the knowledge related to the modeling task has already being gathered. Few of the knowledge still has to gained by means of literatures study and user discussions. Most of the gaps pertaining to the special operation categories. Very less work has been done to model the special operation (e.g. para-dropping, marine diving, terrorist attack, hostages, counter-insurgency commando based special operations etc.)

The main characteristics which are inherited from the past systems and combined into the current systems are as follows:

- Models are at National level: the model will be able to depict Zones (N/S/E/W) level ops with resolution upto Division (One level down to zone level) level.
 - Four characteristics will be prominent (FRMR: Flexibility, Resolution, Maneuver and Resource Allocation)
 - b. FLEXIBILITY: irrespective of map resolution and structure of the segments etc.
 - RESOLUTION: Models will be flexible to vary at different level of resolution.
 - MANEUVERABILITY: maneuvering of the forces in coordination with information, space intelligence for faster decision making and situational awareness
 - e. RESOURCE ALLOCATION: Adaptive Resource allocation: to find the optimal strategies through AI or OR techniques.

- Simultaneous Ops: The model will be able to handle simultaneous Ops by upto 3:1 ratio of Attacker and Defender.
- 3. Types of Ops: The system will be able to handle operations pertaining to Land, Sea, Air, Space, Cyber, IEW (Intelligence Early Warning), Op Lgs, Amph Ops, Maritime Air ops & SF, etc.
- 4. 21/2 Front: The system will be able to handle the scenarios pertaining to the forces like neighboring countries and their allied forces.
- Strategic Inputs and Effects: Inputs of Cyber, Space and IEW etc to be fed textually. The direct and indirect of these factors in terms of delay or attrition will be shown in the results.
- 6. Game Canvas will be used for Preparation, evolution, planning and execution of Ops and lgs plan.

Reference Map:

Table 5. This table shows the past references which are influencing the various modeling processes of large-scale gaming system in the recent times.

	Movement	Detection	Engagement	Replenishment	C2	Attrition	Data	Weightages	Heterogeneity
Air Components		[10]	[10]		[1]	[9-10]		[4],[9]	
Naval Components			[9]		[1]		[7]	[6]	[2-5]
Army Components	[6]	[12]			[2]	[2]	[2]	[24]	
Amphibious Components	[6]		[9]	[11]		[13]	[13]		
Air-TO-Land		[7,12]		[11]				[6]	
Marine-Air-To-Land	[6]			[11]		[3]	[7]		[3]
Cyber Operations						[8]	[14]	[9]	[14]
Information Operations	[15]	[16]	[16]	[8]					
Special Operations						, and the second			

[1] (Allen, 1997),[2] (B.C. Graham, June 1979),[3] (Barnes & Fulford, 2009),[4] (Bourdon, 2014),[5] (Caldwell, Hartman, Parry, & Washburn, 2000),[6] (D. S. Alberts, 1999),[7] (Dupuy, 1995),[8] (Eric Heginbotham, 2015),[9] (F. Bolderheij, 2005),[10] (G.M. Clark, 1969),[11] (H. Brackney, 1959),[12] (H.K. Weiss, 1975),[13] (Hillestad, Bennett, & Moore, 1996),[14] (Hillestad, Bennett, & Moore, 1996),[15] (Hughes, 1993),[16] (Hillestad, Bennett, & Moore, 1996)

3. System Composition (Design Constraints and Methodology)

There are mainly two directions for viewing a system, bottom-up and top-down. For example, the organizational structure of the Indian Railway from the top-down approach can be viewed as Apex Level-Zonal Level-Divisional Level and for bottom-up it is vice-versa. The lower entity in this system is a train and the higher entities is the Chief Engineer of a zone (Northern or Eastern Railway). For developing a management system for such system both the approaches can be followed provided that the totality of the system is maintained

3.1. MBSC

Model Based System Composition
(MBSC) is the emerging technique of system
development that assist in developing system's
architectures based on the modeling
requirements. This architecture is used for

System Composition (Design Constraints and Methodology)

designing, analyzing and testing the system. Like other systems developing techniques this also has four views:

System Conceptualization

- Problem or Operational
 View: problem that system will solve
- Solution or System View: The system itself
- Logical View: Conceptual view of the system
- Physical View: Actual physical view of the system

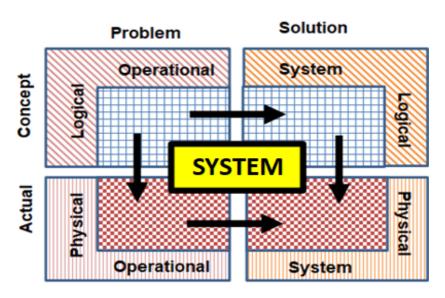


Figure 6.Problem-Solution, Logical-Physical Views of the system

Table 6. Different views of representing System's Complexities using model-based system composition

Problem View			Cal. 4: a. V! a	System Complexity				
			Solution View	Logical	Physical			
•	User's Perspective	•	Behaviour of the	Changes	Required	Changes		
			System	over time	when	technology		
•	Processes	٠	Structure		changes			
•	Objective	٠	Data Flow between					
			Components					
•	Organizational	٠	Allocation of					
	Structure		Functionality					
•	Usecase	٠	Deployment in					
			Real System					
•	Information Flow	•	Alternative					
			Solution					
		٠	Analysis					

MBSC technique can be implemented using Composer Toolbox which was System introduced in the starting version of MATLAB 2 0 2 0 a for developing system architecture. Here we present the Modelling challenges for developing the complex System in a very structured form so that we can manage it scientifically in future. We present here how we can capture the system's architecture intuitively in the digital form, how we can create a digital thread across the different artifacts that we use throughout our development to provide the traceability between cycle requirement architecture and the design, how we can connect our architecture environment or the system architecture with the design environment to do the system-level development. While doing it, we will be utilizing the features of model-based design, which includes the multi-domain experts (math, stat, OR, Comp, military science) to do the component as well as

system level analysis. Before going to the detail, we will first define what is System?

- System
- system is a group of interacting and interrelated entities (these entities may be abstracted military operations converted into mathematical/computer models for representing military Operations in different domains of Land, Air, Navy, Amphibious, Collaborative-Air-to-Land Ops (CALO), Marine-Air-To-Land Ops (MALO), Space-based, cyber-based and Special types. These entities have the defined boundaries, structure and purposes. The system with a defined boundary interacts with its surroundings.
- Let us consider a Gaming Software as a system. It
 is having multiple processing components inside that,
 like movement, detection, engagement, attrition et
 cetera, all them working together, interrelated with

each other, creating one system we call \mathbf{Gaming} System.

System's Complexity

- When we are talking about the system, the first thing that comes in our mind is the system complexity, for example Gaming Software itself is a very complex system.
- Let us take a small example and understand, how the system complexity can be handled with the MBSE concept. Let's take the example of simulating the Encounter Crossing Scenario in a gaming System.
- When the Scenario was first modelled, there was no restriction about the resolutions of the scenario like whether it has to be manageable at Brigade level or at Division level whether it can be manageable for day or night conditions. There were no constraints involved inside that. People start asking

about it, like if it is managed at Brigade level how much time is going to be take or what will be the degree of attrition if it is managed at Division level?

- Then the system designer came up with a fixed resolution, fixed environmental condition which was a solution for that simulation-setup, but not for the different resolutions that can come from the player's side.
- Then the system analyst came up with a solution of having an aggregated model. These models usually were taking minimum number of input parameters.
- There was a component involved now in the system but was not that complex. But then the stakeholder's need, which is nothing but the users, their need keeps increasing. And they said that we want to control how much the resolution they will

decide to the operation to be executed. And then the military domain expert came into the picture, implemented a few components inside that, like aggregated detection, aggregated engagement model, et cetera, and created results for specific resolution.

- Then the user's requirement improved or increased. And they said, I want to automatically control that, or I want to control it with a button. And there comes the computer engineer with his algorithms and blocks of codes, buttons, et cetera, so that we can get a computer algorithm for aggregation. The requirement keeps on increasing. And we know, by looking at the wargaming model, we see just a computer program. But in the background, it might be as complex as this, even more complex than that.
- This is a simple representation of a gaming model, which is having mathematical component, computer

component, military decision-making component, as well as a huge part of the software controlling the system. So, one single resolution, if I want to represent now, I need to create the whole architecture to show what all components existence in that particular system. And how they are interacting with each other, how they are impacting the behavior of the other components.

3.2. Requirements Identification

As we have seen that the system complexity keeps on increasing and system engineering help us in keeping up with that. And this is what the definition of the system engineering is. System engineering is an interdisciplinary field, it is having multiple domains, people working together, and it focuses on how to design and manage complex systems over their lifecycle. Now why do we go for the system engineering? Is the system engineering is about coping with the complexity? It helps avoid omissions and invalid assumptions.

So even before going for the design, system engineering can help us in understanding what all requirements are feasible. We do a lot of the creative studies and understand what all requirements are conflicting with each other, and then take the right steps for that. And this is what is visible. System engineering is not new. It's a very old concept. And

we can see, even Einstein in his time mentioned that if he has a problem, he would have spent 55 minutes in understanding the problem, thinking about the problem, and just five minutes in thinking about the solution. So, this statement emphasized on the time we need to spend in refining the problem statements before even jumping to designing the solution. And system engineering helps us in doing that. It talks about what is the correct way of implementing, or what is the right way of implementing the system engineering.

There are several steps for doing this—

- understand the problem,
- investigate alternative solutions-- that means the creative studies—
- define and agree upon the system architecture, because the multiple stakeholders involved,

- the need to first look how the system will look like, and then agree on that,
- Agree and manage the requirements.

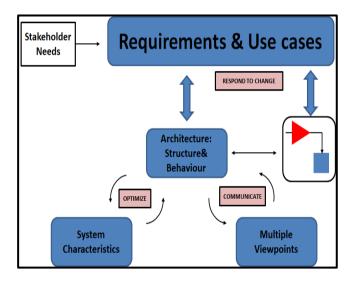


Figure 7. System Engineering Process of the complex system development

First thing that we have to do before we move to any development is to have refined requirements. Agree and manage the interfaces. Because there are multiple domain experts and designers involved, we need to agree to the interfaces and work accordingly. Prepare the test and support system, and track progress against the plan. Now out of these seven, most of these activities we are going to see how we can address to model-based system engineering.

Now let us consider following list of requirements and try to propose a design .The system

- 1. Should be capable for national & strategic decision making
- 2. Should be capable for Theater & Strategic level decision making.
- 3. System for developing SOP of integratd operation.
- Should be able to capture the effect of operations of one service on the operations of the others.
- Should be able to capture the effect of Non-Conventional Actors.
- Should be able to identify the focused Analysis requirement of Integrated operations.
- Should be able to identify Resource and Logistic Requirements.

3.3. Components & Properties

Now before moving to the practical example of implementing the model-based system engineering, let's see what is the system engineering workflow, the standard workflow that we can look into. As we mentioned, that everything starts with the stakeholder needs. we get a very rough level of requirements. From there, we refine our requirements, put it in a more formal way so that it can be understood by the engineering team. From there, we create the system architecture.

The system architecture can be any type of representation. Most commonly used are the **internal block diagram** where we use the **blocks** to represent the components and the **arrows** to show how these components are interacting with each other or exchanging the data. Now on those components and the architecture, we specify its **properties**. When we say **properties**, like if we have a

component, maybe Integrated Air Force (IAF). So, what is the strength of the force? What is the combat power, combat Potential, performance, mission effectiveness? These types of properties we define on those components and use these properties to optimize our architecture based on the stakeholder need-- effectiveness versus performance or time to engage the opponent or the overall behavior, et cetera. It's something that we can optimize.

This is a highly iterative process. Once we refine the architecture, we share this architecture across the Services or across the service commanders and the Theater Commanders. And for that, we need to create the views. There is a different way the team might expect the architecture to look for. For example, Service Commander from Army wants to see just the Integrated Force of Ground components of the Theater, while the Service Commander from Airforce wants to just look into the

Integrated Force of Air Components of the Theater. So based on that, we need to create the different views and share it across the Services, reduce the complexity. And this is a highly collaborative process, as there's multiple forces involved.

integrated task pass these Force we Components to the Service Commander or the Theater Commander, they start analyzing the behavior of those components. At the architectural level, components are nothing but empty boxes, which are representing a component. But at a design phase, we start implementing the behavior of those components, which need to be linked with the architecture so that we can do the complete system level simulation. And when we are doing that, our requirements have moved now from the system level requirement to the design level requirement. So, we have the broader requirements now covering both architecture and the design. And these-- requirement, architecture, and the design-- should be connected with each other so that we can respond to the changes. If something is changed in the requirement, we should be quickly able to understand what will be the impact on the architecture and on the design and take the appropriate action for that. And at the end, we are going to deliver a lot of the artifacts out of it, which include specification, interface, control documents, support code, and multiple more.

So, this is the generic workflow of system engineering or system development process, what we can implement in our basic day-to-day generic problem statement and can adopt it. At the initial phases of the development includes the architecture design, the different type of representation of our architecture, our requirement, management, et cetera. While the later phase, we start doing the implementation in the model-based design.

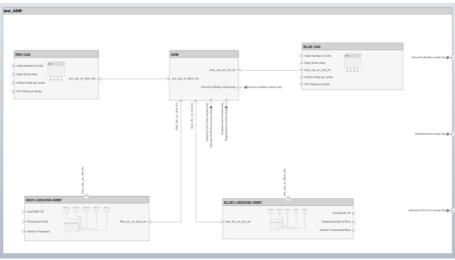


Figure 8. Different Components in a system architecture of an Adaptive dynamic system for representing ground-to-ground combat with close air support.

Components & Properties

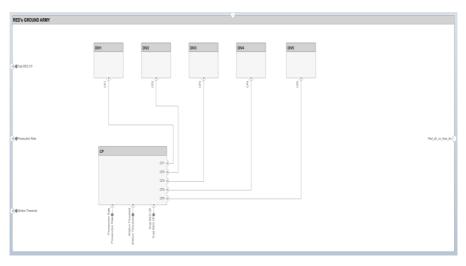


Figure 9. Properties of the RED's ground Army along with linkages.

3.4. Architectures & linkages

The System Composer is a product that is available in MATLAB 2022a, it helps in bridging the gap between the model-based system engineering and model-based design. We spoke about that bridge that we are going to build that bridge. But what goes in that bridge? What all activities we have to perform, we have a defined bridge for that? We will discuss how to develop this bridge using system composer, we start with the concept level, go all the way to the design. First thing what we want is that the tool should be intuitive to capture. It should be intuitive to capture the architecture in the tool and all its components. It should enable the implementation.

At the same time, the tool should allow us to create a digital thread between the architecture requirement and the design. Along with that, tool should allow or facilitate us to

do the analysis at the architecture level so that we can do the different types of analysis. What type of system architecture will be better for our system, providing the effectiveness versus performance criteria given by the user? So that type of the analysis we can do. And we should be able to tackle the complexity inside our system, because our systems are now the system of system, which includes software, model, concepts of operational Services all working together, how we can cope with that complexity.

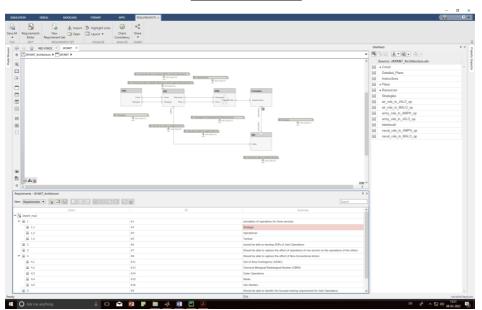


Figure 10. System composer view of the gaming system at the first level

Architectures & linkages

The System Composer, MATLAB 2022a is used to technically capture our system architecture. It helps in doing architecture-level analysis. We can create the views out of that architecture so that it can help in sharing our thoughts with the different Services. It can connect seamlessly with the simulation tool like Simulink. So, it allows us to do the architecture-level simulation, not just the design simulation. It connects with the System requirements, which allow us to create a digital thread across the different artifacts during the development.

We use the System Composer for defining their system architecture. So that the developer will have the access to the entire architecture while implementing a model and can relate to how the requirements are affecting the entire system—i.e., the linkages. If we have a digital link established.

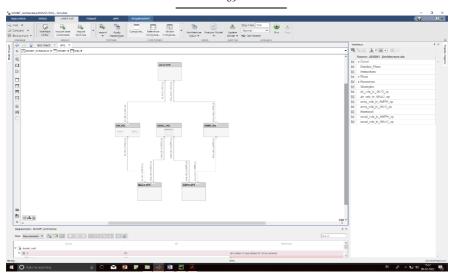


Figure 11. System composer view of the system at the intermediate level

Architectures & linkages

we change a requirement; we know how it's going to impact the overall architecture. At the same time, component commander will know how its components is going to impact the overall system.

Now let's take an example and see how we can implement the model-based system engineering in any Problem. As we have seen there are the different stages of development cycle. It starts with stakeholder needs, goes to requirement, then we create an architecture, then we define the system characteristics and do the creative study. We create views to share with the domain experts. And then we implement the algorithm of the component in computer programs (e.g. Simulink). What we are going to do is that we are going to take a simple example of a Military Operation Scenario. We are going to design a Gaming System, or the architecture of a Gaming System, which is expected to help in planning in such military operation scenario. Whatever Military

Operation is planned involved with integrated Service domain can be planned in the desired Gaming System. And we can understand that the system will capture all the components involved in an integrated environment.

So, we can see that, everything is tied to the stakeholder need. One user may ask; the system should be capable to incorporate the defense of **cyber** as well as **space** components in an integrated military services operation. There is an aggregated force modelling requirement. Another user came and said that the system should be able to analyses the strategic and operational decision-making problem in a very quickly and timely manner. So that there is a requirement of the model with simplified with less parameters.

So, like this, we will be having a multiple type of requirements or **design constraints**, like modelling

resolutions should be Division, Task group and half squadron for Army, Navy and Air forces respectively. It should be able to capture the Simulation of all Services, and so on. From there, we start defining the more formal requirements. From there, we defined the architecture for the system. In our system at a very high level, we understand that there is going to be a Collaborative Gaming System. We represent that as a block. And as a Theatre coverage area is very large, the Commander is not going to manage all the lower entities below the defined resolutions. To control that, the theatre Commander is going to communicate with the CDS and IDS. There is another block for that, that is SHOs (Service Head Quarters). All possible operations including homogeneous as well as heterogeneous at the desired resolutions will be captured in this block.

Now then we define how these components are going to interact with each other, whether the CDS & IDS will send

some command to **SHQs** and receive some plans from the Individual Service Hq. All these things we start defining in the architecture behavior. Then we start connecting these architecture components with the requirements because we need to represent which component is satisfying which requirement from the stakeholder. Now when we are defining an architecture, we have multiple options available in the domain. There are a number of command-and-control architectures available in the domain that we can use to design our System.

So which architecture is fitting into our stakeholder need? we start defining those properties of those architecture which we can get from the Services on those components and start doing the analysis of those architectures to understand which is fitting into our need. And as we mentioned, usually the architecture tends to become very complex. But for example, if I'm a theatre Commander. I want to see the architecture. I don't want to see all these things which is having a mathematical component, engineering components, environmental components, et cetera.

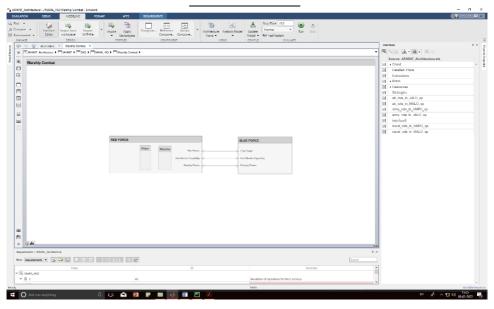


Figure 12. System composer view of the system at the deeper level

Architectures & linkages

We want to create a more filtered view of the components which are relevant for us so that we can read it and understand it. We can create a view which is relevant to the software engineer, or a view which is relevant to the system engineer for the physical view so that they can focus only on their part, or maybe the different components which are falling into the different views. For example, on Div Hq, which is a physical component but also is a part of a system view because it is having an algorithm running inside that. But once all this thing is done, we share these views with the different teams, those teams start developing the algorithms for those components.

So, we connect the behavior of those components with the architecture. Now having said that, the different activities that we mentioned starting from requirement, architecture, trade studies, integration, et cetera, a question we faced here is that which all activity of model-based system engineering

we are working on or we are planning to work on. Whether we focus on the system requirement or the requirement engineering, or we also work on the architecture design, architecture-level analysis trade studies, or we also go all the way till the complete system level simulation and analysts.

A11 these activities in Simulink are done environment with an integrated view and integrated tool chain. Now we can define our requirement in any tool. It might be Word; it might be Excel sheet where we want to use it. Simulink also provide the same environment, what we call the Simulink requirements. In Simulink requirements, we can import our requirement in our design environment, which is Simulink and System Composer, or we can alter our own requirements or we can edit all the important requirements. Everything can be done in the single environment. we can find in an interface of Simulink requirements. If we go, we have a hierarchical requirement where we are showing the stakeholder need of the System to plan inter service operation, as well as we captured that it should fit into the light weight mathematical computing package.

From there, we have the System Characteristics, like the System should be able to plan an integrated operation participated by all services. It should be of strategic or operational level planning. So, whatever the requirement to capture in any tool or in any format we can capture it inside the Simulink requirements or import it inside the Simulink requirements. Why we are doing it? So that we can have requirement and design and architecture in the single environment. What is the benefit of it? So, once we have the requirements ready, whether it is in our tool, a third-party tool, or in the Simulink environment, the next phase of the activity is defining the actual architecture.

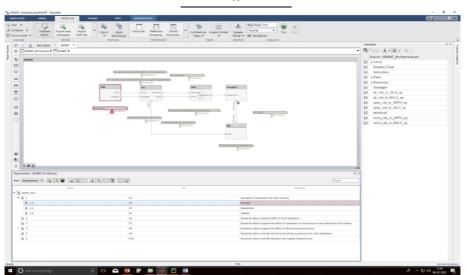


Figure 13. Requirement analysis of the system

Architectures & linkages

Now at very high level, as we have mentioned that we have three components -- CDS, IDS, and SHOs. And Formation and IND interacting with each other. How are we going to represent that part, represent that in the architecture? So, the environment that we see is called System Composer. What do we have to do is just thread and create a box which represent our Formation and IND, as well as our CDS, IDS and SHO? Then we will specify that there will be one command moving from SHQ to Formation and IND. Let's called those commands. And then the FORMATION and IND will be sending the scenario specific operational data back to the Command Centers. It is as simple as that.

Now we can go inside the SHQ and start designing the operations inside the Army Hq, Naval Hq and AirForce Hq. It is a hierarchical diagram. We can go deep inside each and every component to start designing the architecture for that. At very high level, this is how our overall system will look

like. It is as simple as creating box and connecting the Commands. When we are doing that, these boxes are nothing but what we call components. These are called ports, which are the entry and exit point from those components. And these are the connectors, which represent there is a message or the command of information flowing from one component to the other component. Now we start creating this architecture. Assume that you define the architecture in detail. We can go inside the Army Hq and start defining that there is a Armour Division, there's a Mechanized inf Div, there's an Artillery Regiment, there's a SFF Battalion which is having a special operation Troop, and so on. All those components We can start defining in the Army Hq.

The next phase is to represent which component we added for which requirements. Or which component is satisfying which requirement? And for that, what we do is what we call requirement allocation. So, we import the requirement inside the Simulink environment. And that exactly we can see just below the editing tool. What we have to do is just drag a requirement and drop it on any of the components. This activity creates a bidirectional traceability between the components inside our design and the requirements. Now these two are connected. And that's how we can create an allocation between the requirement and the architecture. Now in this case, if we change a requirement, it will immediately highlight a component inside an architecture and will tell us this is the component which will be impacted because of the requirement change.

So, if we see an overall view here, through this we can press the system requirement with the architecture using the Simulink requirement tool. At the bottom, we can see the requirements right there just below the editing canvas. And the moment we connect the requirement with the architecture component, we can see that requirement is now visible on

the editing canvas with the arrow linking the requirement with the component. Now it is easy for us to review, to audit this whole architecture, to understand which component is implemented for which Services, or which requirement is implemented by which component-- multi-ways. Now once that is done, the next phase of any architecture design is defining the interfaces. So, what do we mean by defining the interfaces? So, in the beginning, we have mentioned that send command to the CDS/IDS will SHO. Formation/IND will send the operational information back to the CC, which, to represent the overall system and how the components are interacting with each other, is sufficient enough.

Suppose we want to design a military operation, for example, to an Air Support to Ground Force and tell the Army HQ and Air HQ, integrated Task Force is supposed to get the commands. How the Integrated Task Force

Commander will understand, what do we mean by command? What all goes in this word command. And that is where we have to specify the interface information on each and every port so that it can go to the Integrated Task Force Commander and even for the domain expert can read it and understand what do we mean by command. And how can we do that? It is as simple as that-- creating a bus object in Simulink. We created one variable or one object create name command. We selected a port and assigned that command interface to that. So, the moment We click on the port, interface is highlighted, and vice versa. Now in that command, we can create the members or the components in that. For example, our ground Formation/IND will have Attack, Defend, and, let's say, prosecute/withdraw commands going to the CC.

The moment We do that, now for each and every of this attack, defend, and withdraw, we can see we can specify

what is its duration, what is its rate. What is the minimum/maximum rates and duration? What is the description of this command? So, what happens is that when we pass this system's component to the software engineer, that software engineer will understand that we are going to get three commands from the CC. These are the data types and the specifications of the commands. And now we can start designing the behavior accordingly. So, interface information is very important, very critical part of the architecture designing. Because when we get these components from the different designers and try to connect it together as an integration engineer, most of the cases we find an issue because the interfaces are not maintained properly or are not synchronized with each other.

And these are the interface control document can help us in doing it. Now inside the System Composer, we can define these components and these interfaces. And these interfaces as it is, will be used inside the Simulink, because they use the same bus objects. So, there will not be any mismatch between the design phase and the architecture phase. So, as we mentioned, we can create the interfaces. Now once we create the interfaces, our architecture design is ready, the interfaces are ready. Now we go to a stage one step forward, and then start defining more **properties on the architecture**. We can start capturing the system characteristics and properties. Now on our architecture, we have different components. For example, we have a CDS/IDS/SHQ.

Now this system has some cost in terms of decision making or planning. This system has some measure of effectiveness. This system has some combat potential, combat power, measure of performance (MOP), measures of effectiveness (MOE) etcetera. We should be able to capture those properties on the system component or

on the force structuring or on the package formation which we used inside the architecture.

Let's see how we can do that. So, if we see here, in this case, we created three different architectures. Why we created three different architectures? Because we have three different Military Operation System conceptualized. One is for Command, Control and Communication. One is for Resource Management. Or we can use an architecture focused on the attrition process. Now we have three different architectures. Any one of this is something that we want to take forward for the design phase, we don't want to do the designing for all three. Because at the end of the day, we going to use one of it. So, we need to do the analysis that which system architecture is going to fit into my stakeholder needs. For doing that, we need to define a few properties on the architecture.

So, when we said properties on the architecture, what do we mean by that? We can define a stereotype. For example, we select an architecture. On the right-hand side, we can see there are multiple properties captured for that-- what is **operational Capacity**, what is **planning and evaluation time**, **duration**, **resolution** of the system, total **resources handling**, total **combat potential**, **combat power**, et cetera. Now these properties are static properties. we can get it from open data set. we specify it on the System so that we can use it for further calculation.

We go inside the System architecture and we go to the Combat Potential. Combat Potential for us is nothing but the aggregated combat values of the force. So, if We go further inside, we can see weapons. We have weapons properties. What is the range coverages of the weapons? What is the FOV, operational synchronization of the weapons? As we can imagine, the coverage and the engagement range are

going to define how much area my system is going to capture. We go to all three different architectures and start defining these properties for the components that we are going to use. Whether it is C3ISR based system, or it is some Logistic Management system, or different attrition induced system. We start defining these properties. And why do We define those property? So that We can do the analysis on that particular architecture. So, what we are seeing right now here at the bottom is what we call analysis viewer.

This is an inbuilt feature of the System Composer, which help us in connecting our architecture with MATLAB engine in the background. Use the MATLAB analysis capability to do the trade study or the analysis at my architecture level. So, if we remember, our requirement from the stakeholder was that my system should be able to capture planning for conducting operations with all services. What we are going to do is that we are going to calculate the confidence level of

all three architectures and figure out which is fitting into our requirements.

So, the view that we are seeing here is called the analysis view of the System Architecture. In this, all the components, all the three architectures are defined. If we can see on the left-hand side-- C3ISR based architecture, Logistic based architecture, and attrition-based architecture. This view captured all the different properties inside the architecture, called the MATLAB script in the background, and do the analysis on my architecture. So, when we did that, when we ran our MATLAB analysis on our three different architectures, it calculated the different properties out of it.

And what we figured out is that only the first architecture is having a Confidence level of 95%. Rest two are around 90% and 85%. The moment we did that, we can highlight only the one architecture, one system architecture is fitting

into our stakeholder need of all the Services. So, we can scrap other two architectures and can take the first architecture, which is C3ISR based architecture for the further detailed design, high fidelity design of those components and pass it to the design teams. So, at architecture level, we can quickly do a trade study and analysis and figure out which one we want to go forward with.

Now once we do that, we need to pass this architecture to the design team. So, when we say we need to pass the architecture to the design team, the whole architecture will not go to one team. This architecture has few operational components, will go to the software and mathematical modeling team. So, we need to create the views. For example, we want to create an architecture view to the modeling team. So, what we can do is that we can just create a filter, then give us a view of all the physical system. The

moment we do that we can create a view. And as we can see this complex architecture will be filtered out but the components, which are having only system properties.

All the components which will go to the either software engineer or the mathematical modeling team. And we filtered out all the relevant components outside that. So not only that, we can create different other views. For example, hierarchical diagram we can create out of the same architecture. On the right-hand side, we can see the hierarchical diagram, which represent my same architecture that I created more into the functional decomposition view. Like in my sensor module, we will have a sensor and a its view of the operational scenario. What was the ports of that sensor and operational view? What are the properties of those components, et cetera, will be visible in the hierarchical view? So, a lot of the people prefer to look into the architecture as a decomposition view. Not only that, we

can use the same architecture to define more behaviors of the architecture or behavior of the system-- for example, a sequence diagram. So, on the right-hand side, what we can see is the sequence diagram of the block diagram that we created on the left-hand side. It represents how the command will be passed from one component to the other component. Or in this case, we call that lifeline. And how the command will be written back, what processing will happen, and so on.

If we create a different lifeline in the sequence diagram, automatically components will be created. If we create communication between the lifelines, it will be automatically created inside our architecture. So, both are synchronized with each other, whether it is a sequence diagram or it is a block diagram that we created. Now what we have done is that we created the filter view. We created the different views and passed it to the design teams. Now design teams are going to specify the behavior of those components. And they

use the Simulink to do that. What we need now is to connect our architecture component with the Simulink behavior. So, we can connect the System Composer with Simulink in model-based design by simply calling the Simulink models behind those components, whether it is a physical component or it is a control or the calculation algorithm mathematical models, and just simulate the whole architecture. How can we do that? We can have a quick view here. So, this is the internal architecture of the System in which we have a component for a communication interface.

What we can do is that in that component we just specify there is the Simulink behavior of that component defined. The moment we define the Simulink behavior, we can see the ports are automatically created because it took it from the Simulink models already. What we have to do is that we have to connect this component with the rest of the architecture as simple as just dragging the signal and dropping it on the

component. If we go to the Simulink design and, for example, if we make any change in the design. That's supposed to create and port here, now for this port, if we remember, we specified the interface information at the architectural level, we can use the same interface information at the Simulink design level. we don't have to recreate the interface so that we there is no mismatch in the interface when we connect the different components. The moment we do that, a new port is created at the architectural level for that component. What we have to do is simply connect it with the rest of the components. And now my Simulink, the architecture, is ready with the Simulink behavior, and we can simulate and analyze the whole architecture at the system level. we don't have to do the component-level testing.

we can do the complete system-level simulation. And the moment that is done, what is the next stage? we start doing the testing of my system. And for that, we are using Simulink Test, which is our test environment, we can link our test scenario with the requirements, we can provide the triggers. Triggers are nothing but the simulation test scenarios to the architecture. And the moment we do that, now we see, whatever we see the graphics here, is actually the complete system simulation which includes the plans. It includes the controller. It includes the communication, everything inside that. Now when we do the complete system-level simulation, we can go and see the behavior of my complete system. So, what we are doing here is that we are just opening the activities of the Army Components. So, if we see here, we have the Army movement of Ground Forces, which is a sort of a typical ground activities.

And just below that, we have the actual Operation Planning Tool, which sort of planning the COAs as we expected. We can look into the results and analyses that what exactly our behavior is, that it is fitting into my requirement

or not. The bottom line here is that we are simulating the complete system and not just one component in that. So, what we have seen so far, this was our initial flow or the workflow for the model-based system engineering. What we have seen is that we can use Simulink requirements to capture our requirements, can use the System Composer to create and define your architecture, use the properties to link it with the MATLAB environment and do the trade study and analysis, create views to share it with the different stakeholders, and, at the end, connect our architecture with Simulink behavior so that we can do the complete systemlevel simulation.

And while doing it, everything is connected with a single digital thread so that any change in our system will directly be captured in the different artifacts and can be seen what will be the impact on the different artifacts for any change in the requirements, architecture, or in the design. Now having

said that, we understood-- we hope we got the basic behavior or basic workflow of system engineering and how we can implement that in System Composer.

3.5. Multiple Views

Multiple views show how the components are deployed and employed to achieve different levels of objectives from the perspective of the system design. Based on these objectives' campaigns are planned and executed. Planning is done by breaking the National Objectives into hierarchy of objectives, i.e., Strategic Objectives, Operational Objectives and Tactical Objectives.

- National Security Objectives: President decides the national objectives
- National Military Objectives: MoD, CDS decides the military objectives
- Campaign Objectives: Combatant Commander defines campaign Objectives

- Operational Objectives: Combatant and component commander
- Operational Tasks: elements of component staffs define the operational tasks

Strategies links between different levels of objective in the hierarchy. An objective is achieved through implementation of strategies. Meaning of strategies varies with different levels e.g., at the campaign level when the higher commander is thinking about his strategy of assigning weightages of an effort by different components (air, navy, ground) at the same time the Wing Commander or Squadron Commanders are thinking about their strategies of how to execute the operational tasks assigned to them by their higher commander

Processes at the Strategic/Campaign Level

- Plan to support national military objectives
- Plan for certain scenarios

- Broad idea about the availability of forces
- They have Planners and Intelligence Specialists
- Define campaign Objectives
- Weight of effort to air, sea, land components
- Define manoeuvre of ground forces, allocation of air forces, direction of move for naval and amphibious components.
- While defining the campaign objectives it also details the forces available, allies and constraints.
- Detailed Planning-COG, objective, intent, matching force elements to this objective, time-phased employment-deployment of forces.

- Live adjustment of campaign plans, policies, status of forces based on Intelligence Assessment
- Resolving disagreement in effort weights, targeting priority, demand of resources
- Models required to fulfil above processes are:
 - Effectiveness and performance of components are assessed for different air-scenarios like Recce, air-borne assault, air-to-land, special heli-borne, air transport operations.
 - Models for evaluating capabilities of available forces.
 - Model for time-phased employment & deployment of forces
 - Model for detailed planning-COG analysis

- Battle management –target prioritization, resolve disputes.
- Resource movement

Processes at Operational Level

- Employment-deployment of air power
- Weights of effort at operational level
- Allocate air force accordingly
- Adjust weights and allocation as battle progress
- Detailed plans-tasking order (targeting, suppression, refuelling, engagement)

Processes at Engagement Level

- Engagement monitoring by AWACS
- AWACS directs fighter to interceptors

- AWACS aids attacking aircraft with ground targets
- Weather monitoring with the attack
- Target selection (aim point, target status), route selection (ingress-egress, awaking AD, radars), decide on tactics (make use of environment, target properties, weather condition, selecting type of weapon)

Processes at all levels

- Intelligence Gathering of Adversaries: performance of weapon system, tactics, intention
- Intelligence Assessment: Situational Awareness
- Training: Wargaming
- Communications: two-way communications

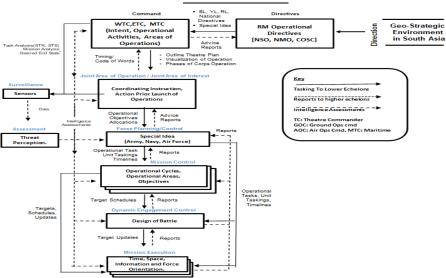


Figure 14. Block diagram of the System's Architecture $\frac{}{}$ System Design

The system view from the Army commander point of view is based on one of the theater levels measures of effectiveness of FEBA or FTL or FUP (Forward Edge of Battle Area or forward troop line or forming up place). This model is divided into four sub models. These are as follow:

Theatre Control

Resource Allocation

RA=f (loss (wpn, pers), Mission Effectiveness, demand)

Initial Deployment

Deployment=f (threat, availability, effectiveness)

Reinforce

R=f (loss (wpn, pers), Mission Effectiveness, demand)

Combat Unit Densities per area

Ground Combat

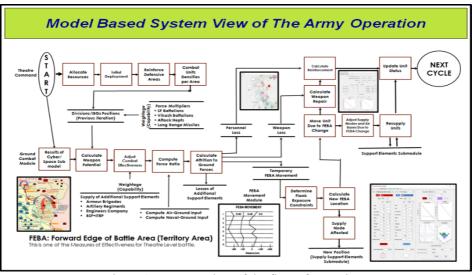


Figure 15. System view of the flow of operations

System Design

- Weapon performance/weapon effectiveness
- Unit Combat effectiveness
- Force Correlation (Air + Naval)
- Force Exchange (additional Support Elements)

FEBA Movement

- Flank Determination
- New FEBA position
- Changes in the Supply Node
- New Position

Status Update

- Loss Recording
- Reinforcement Calculation
- Update Unit Status

 Effect of FEBA movement (Position, supply mode, Air bases)

Following figure shows the system view of the flow of the operation.

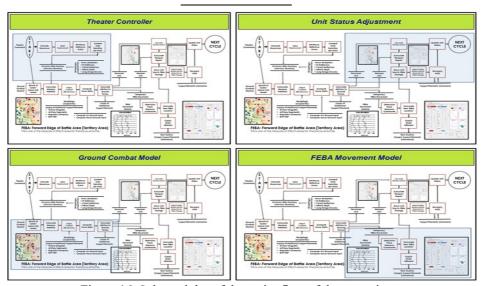


Figure 16. Sub-modules of the entire flow of the operation

Multiple Views

A Joint Task Force (JTF) is a temporary grouping of combat units (assuming at the abstractions level of Army Divisions, Air Half squadrons and Naval Task Force), under one higher commander, formed for the purpose of carrying out a specific operation or mission. It may be a semi-permanent organization of units under theater commander, formed for the purpose of carrying out a specific strategic operation. The JTF is considered as a combination of Forces and Command, Control & Communication (C3) system used within collaborative operations.

This system considers nine major modules, and they are (i) Air (ii) Navy (iii) Land and (iv) Amphibious, (v) Collaborative-Air-Land Operation (JALO) (vi) Marine Air-to-Land Operations (MALO) (vii) Cyber (viii) Information (ix) Special (see following figure). These modules simulate the functioning of the sub-systems as mentioned above. The Data sub-module will keep

3.6. Categorization of the System

The major components considered for the Collaborative Operation Scenarios are illustrated in following figure.

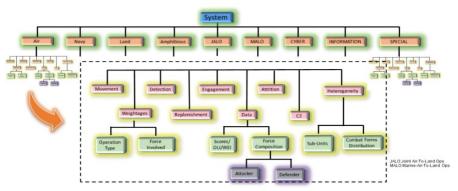


Figure 17. Major sub-systems of the Collaborative Gaming System

Categorization of the System

performances of systems in the form of scores or WEIs, whereas Weightages sub-module will store all the information about the operation type and force composition. The Heterogeneity sub-module will consider both sub-unit's information and Combat form distribution. These modules will contain mathematical models, databases, operational philosophy translated into algorithms and certain decision rules to represent theater level systems/sub-systems.

The designing concept of these nine modules are discussed in the subsequent books. The numbers placed in the right-down corner of the blocks indicate corresponding books.



Figure 18. Decomposition of the system

Each subsystem can be further classified into nine groups as- movement, detection, Engagement, Attrition, Heterogeneity, Weightages, Replenishment, Data and C2. These sub-systems will be integrated in part or in simulation Operational Scenarios both in event-or time- driven set-up.

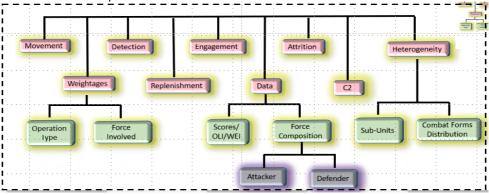


Figure 19.Decomposition of major sub-systems

3.7. Conclusions

In the overall perspective of model-based system Design (MSD) approach we can infer some of its important aspects as follows:

- 1. Unifying the Interfaces and not collecting properties.
- 2. Manifesting the main Capabilities.
- Satisfying individual requirements of the basic components so that the overall system's requirements can be achieved.
- 4. Realizing the autonomous properties of components.
- 5. Equipping self-responsiveness.
- Training data for supporting each component's behavior.
- 7. Minimizing unnecessary properties
- 8. Inculcating basic model implementation.
- 9. Developing model historically
- 10. Having simple model in its basic core.

4. Appendix 1: Force Ratio Attrition Model

In this type of model, the bigger units that are considered, such as Division and Brigade, each element of them are combined to create a single Measure. This procedure is done for both the defender and the attacker. Then they are modeled on their ratio and conclusions are drawn on how many damages will be caused on both sides. The three terms are very familiar in this regard. These are the **Fire Power Score**, Fire Power Index, and Force Ratio. The score is a relative numerical value for all weapons. The FPI is the aggregated value and the FPR is the ratio of attacker and defender. There are some limitations of this approach, such as the linear additive nature for finding an index for aggregated forces. An example of a force ratio model is the ATLAS ground attrition model.

The firepower score approach measures the combat power of a unit by summing the combat power values for each weapon system in the unit. Suppose that there are n different types of weapon system in a combat unit and that:

 X_i = number of weapons of type i in the unit (i = 1, 2... n).

Define the firepower score for each weapon of type i to be

 S_i = score value representing combat power for each type i weapon, and define the firepower index of the aggregated unit to be

$$FPI = \sum_{i=1}^{n} X_i \times S_i \tag{1}$$

If FPI(A) and FPI(D) represent the total fire power index of the attacking and defending force respectively then force ratio is defined as:

$$FR = \frac{FPI(A)}{FPI(D)} \tag{2}$$

For area weapons (artillery) the firepower score was defined as:

$$Si = (daily ammunition expenditure) * (lethal area per round) (3)$$

For point fire weapons the score was similarly defined as:

$$Si =$$
 (daily ammunition expenditure) * (probability of kill) (4)

5. Appendix II: Lanchester's Models

Lanchester's models are suitable for developing theaterlevel campaign model using High Resolution Simulation data. In reality, actual historical combat data is not easily available and common practice is to develop High Resolution simulation data with detailed design. Various literatures have demonstrated that estimating attrition rates from high-resolution simulation and using Lanchester's model for linking the various resolutions of different simulation models. Lanchester's models are flexible for both homogeneous as well as heterogeneous situations. Lanchester's models are used for theoretically consistent force aggregation and disaggregation. Defence planners and decision makers use Lanchester's models to predict likely outcomes of combat processes.

For the case of two homogeneous forces, the following Lanchester's-type equations are generally used in the highresolution simulation as given in following equation.

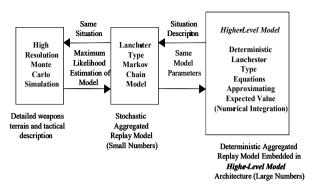


Figure 20. Hierarchy-of-modelling concept

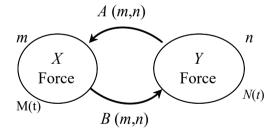


Figure 21. Schematic diagram of combat interactions for stochastic battle corresponding to the deterministic Lanchester's type equations.

$$\begin{cases} \frac{dx}{dt} = -A(x, y) & \text{with } x(0) = x_0 \\ \frac{dy}{dt} = -B(x, y) & \text{with } y(0) = y_0 \end{cases}$$
 (5)

where x(t) and y(t) denote, respectively, the X and Y force levels at time t. Let us assume that there is no replacement or withdrawals and A and B are the attrition rates of the X and Y forces, respectively. If we want to statistically estimate the attrition rates in the model (following equation) from simulation output data we must consider a stochastic version of the model in which casualties occur randomly over time. Letting M(t), a random variable, denote the integral number of X combatants alive at time t (with corresponding realization denoted as m) and similarly for N(t) which pertains to the Y force as shown in the above figure, we then have the following Kolmogorov equations for the evaluation of the state probability for 0≤m≤m0 and 0≤n≤n0 as given in above Equation.

$$\frac{d}{dt}P(t,m,n) = A(m+1,n)P(t,m+1,n)$$
+ (6)

$$B(m, n + 1)P(t, m, n + 1)$$

- $\{A(m, n)$
+ $B(m, n)\}P(t, m, n)$

where $P(t,m,n)=P[M(t)=m, N(t)=n \mid M(0)=m_0, N(0)=n_0]$

Now, suppose attrition rates appear linearly in A(m, n) and B(m, n), as given in following equation

where $A(m, n) = ag_a(m, n)$ and $B(m, n) = bg_b(m, n)$ (8)

where
$$g_a(m, n) = \{1 - (1 - P_A)^m\} n \text{ and } g_b(m, n) = \{1 - (1 - P_B)^n\} m$$
 (9)

 P_A and P_B refer to the probability of target availability of X and Y respectively; and a and b are Lanchester attrition-rate coefficients and are constants.

Abbreviations

AAM Air-To-Air Missiles AAR Air to Air Refuelling

AB Air Base

ADA Aeronautical Development Agency
ADIZ Air Defense Identification Zone
ADCC Air Defense Control Center
ADDC Air Defense Direction Center
AEW Airborne Early Warning

AGPL Actual Ground Position Line
AIP Air Independent Propulsion
ALCM Air Launched Cruise Missiles
AMD Anti-Missile Defence

AMRAAM Advanced Medium Range Air-to-Air Missile

AOA Amphibious Obiective Area

ARTRAC Army Training Command
ASBM Anti-ship Ballistic Missiles
ASM Air-to-surface Missiles
ASV Anti-surface Vehicle
ATGM Anti-tank Guided Missiles

ATCAL Attrition-Calibration
AWACS Airborne Warning And Control System

BAS Battlefield Air Strikes
BAI Battlefield Air Interdiction
BFSRS Battlefield Surveillance Radars

BVR Beyond Visual Range
CB Counter-Bombardment

CBMs Confidence-Building Measures
CC Component Commander

CCS Cabinet Committee on Security

CDS Chief Defence Staff

CEP Circular Error Probable

CIA Central Intelligence Agency (US)
CIDS Chief of Integrated Defence Staffs
CNAN Capabilities for the Navy After Next

COSC Chiefs of Staff Committee

 C^3I Command Control Communication and

Intelligence

 C^4I^2 Command, Control, Communication, Computers,

Intelligence and Interoperability

CSG Combat Strike Group

CSRS Counter Surveillance Reconnaissance System

Defense Advanced Research Project Agency DARPA DF Direction Finding

DCA Defensive Counter-Air

DSSC Defense Services Staff Collage EEZ. Exclusive Economic Zone

ETI Exhaust Trail Indicator FAC Forward Air Controller FGA Fighter Ground Attack FOFA Follow On Forces Attack

FARP Forward Armament Refueling Point

FGA Fighter Ground Attack FUP Forming Up Place

JTLS-Go Joint Theater Level Simulation-Global Operations

JOC Joint Operations Center HAT High Anti-Tank HGV Hypersonic Glide Vehicles **IEW** Intelligence Electronic Warfare

Inter Continent Range Ballistic Missiles **IRBM**

ISI Inter-Services Intelligence

LAT Light Anti-Tank

LCAC Landing Craft Air Cushion

Low Intensity Conflict Operations LICO

LPD Landing Platform Dock LPH Landing Platform Helicopter

LGB Laser Guided Bomb LRF Laser Range Finder

LST Landing Ship Tanks

MAD Magnetic Anomaly Detector

MEU Marine Expeditionary Units
MHWS Mean High Water Springs
MLWS Mean Low Water Springs
MPA Marine Patrol Aircraft
MOP Mobile Observation Post
MRBM Medium Range Ballistic Missiles

NDA National Defence Academy

NDC National Defence Collage NSA National Security Authority or Advisor NSAB National Security Advisory Board

NSC National Security: Council

NWDC Navy Warfare Development Command

OCA Offensive Counter Air
OTH Over-the-horizon
PAF Pakistan Air Force

PGM Precision Guided Munition
PNT Position, navigation and timing

RCL Recoil-less Launcher RFP Request for Proposal

SAGE Sequential Analytic Game Evaluation

SAGW Surface-to-air guided weapons SLOCs Sea Lanes of Communications Short Range Ballistic Missiles SRBM SSBM Submarine Ballistic Nuclear SSGN Submarine Attack Nuclear SSM Surface-to-Surface Missiles STSB Ship-To-Shore Bombardment TLC Theater-Level Campaign

Glossary

Remarks

CSRS The Counter Surveillance Reconnaissance System (CSRS) is a mobile platform that could temporarily deny enemy satellite's ability to collect information

Differential Equation (DE)

Term

- 1. DE is equation that contain an unknown function and some of its derivatives.
 - 2. **Order** of a DE is the order of highest derivative, degree is the **degree of the highest order derivative** in a DE.
- 3. First-order linear differential Equation

$$\frac{dy}{dx} + P(x)y = Q(x)$$

4. Integrating Factor for solving FDE

$$I(x) = e^{\int P(x)dx}$$

5. Homogeneous Differential Equation M(x,y)dx + N(x,y)dy = 0

For Lanchester's model

$$\frac{dx}{dt} = aX^p Y^q$$

$$\frac{dy}{dt} = bY^p X^q$$

Hence
$$\frac{dy}{dx} = \frac{by^p x^q}{ax^p y^q} = \left(\frac{b}{a}\right) \left(\frac{y}{x}\right)^{p-q} \text{ if } v = \frac{y}{x}, vx$$

$$= y, v + x \frac{dv}{dx} = \frac{dy}{dx} = f(v)$$

$$\therefore v + x \frac{dv}{dx} = f(v)$$

$$v + x \frac{dv}{dx} = f(v)$$

$$\frac{dx}{dx} = \frac{dv}{f(v) - v}$$

$$\int \frac{dv}{f(v) - v} = \int \frac{dx}{x}$$

$$\int \frac{dv}{\left(\frac{b}{a}\right)v^{p-q} - v} = \int \frac{dx}{x}$$

$$\int \frac{adv}{bv^{p-q} - av} = \int \frac{dx}{x}$$

$$\int \frac{adv}{v(bv^{p-q-1} - a)} = \int \frac{dx}{x}$$
$$\int \left[\frac{bv^{p-q-2}}{bv^{p-q-1} - a} - \frac{1}{v} \right] dv = \int \frac{dx}{x}$$

 $bv^{p-q-1} - a = k$ Now let $\Rightarrow b(p-q-1)v^{p-q-2}dv = dk$

$$-\int \frac{dv}{v} + \int \frac{dk}{k(p-q-1)} = \log x + c$$

$$-logv + \frac{1}{p-q-1}logk = logx + c \tag{1}$$

Substituting
$$x_0$$
 and y_0

$$-log\left(\frac{y_0}{x_0}\right) + \frac{1}{p-q-1}log\left[b\left(\frac{y_0}{x_0}\right)^{p-q-1} - a\right]$$

$$= log x_0 + c$$

Hence solution is

$$c + \log x_0 = -\log\left(\frac{y_0}{x_0}\right) + \frac{1}{p - q - 1}\log\left[b\left(\frac{y_0}{x_0}\right)^{p - q - 1} - a\right]$$

$$= \frac{1}{p-q-1} \left[-(p-q-1)log\left(\frac{y_0}{x_0}\right) + log\left[b\left(\frac{y_0}{x_0}\right)^{p-q-1} - a\right] \right]$$

Term Remarks
$$= \frac{1}{p-q-1} \left[log \left(\frac{x_0}{y_0} \right)^{p-q-1} \right]$$

$$+\log\left[b\left(\frac{y_0}{x_0}\right)^{p-q-1}-a\right]$$

$$= \frac{1}{p-q-1} \left[log \left[b \left(\frac{y_0}{x_0} \right)^{p-q-1} - a \right] \right] \times \left[\left(\frac{x_0}{y_0} \right)^{p-q-1} \right]$$

$$= \frac{1}{p-q-1} \left[log \left[b - a \left(\frac{x_0}{y_0} \right)^{p-q-1} \right] \right]$$

$$= \frac{1}{p-q-1} \log \left[b - a \left(\frac{x_0}{y_0} \right)^{p-q-1} \right]$$

$$\therefore \qquad c = \frac{1}{p-q-1} \log \left[b \right]$$

$$p-q-1$$

$$-a\left(\frac{x_0}{y_0}\right)^{p-q-1} - \log(x_0)$$

$$\therefore c = \frac{1}{p-q-1} \log\left(\frac{b-a\left(\frac{x_0}{y_0}\right)^{p-q-1}}{(x_0)^{p-q-1}}\right)$$

$$c = \frac{1}{p - q - 1} \left[log \left(\frac{b}{x_0^{p - q - 1}} - \frac{a}{y_0^{p - q - 1}} \right) \right]$$
System Design

Term Remarks

Substituting the value of c in equation (1) we get the solution as

$$y^{p-q-1} = \frac{a(xx_0y_0)^{p-q-1}}{b(x_0y_0)^{p-q-1} - b(xy_0)^{p-q-1} + a(x_0x)^{p-q-1}}$$

FORCE

Forces are **elements** that perform *actions* directed against a given enemy element or target. A commander may activate his forces, causing a collection of actions to begin which, in themselves, cause processes that result in some measurable result. The combination of these actions creates an effect on the enemy which is called combat power (Hughes,

1993).

A Group of naval ships that operate and fight together.

JTLS -Go

The Joint Theatre Level Simulation-Global Operations (JTLS-GO) is an interactive, web-enabled, joint and coalition wargaming system. JTLS-GO represents civil-military decision-making environments from a globally integrated operational-level perspective, which includes Air,

Term	Remarks
	Land, Maritime, Space, Intelligence, Logistics, and Special Operations. These environments can be configured and scaled to examine joint tasks, operations, functions, and missions at the strategic national (SN), strategic theatre (ST), operational (OP), and tactical (TA) levels of war. The JTLS-GO is primarily an operational-level simulation.
LCAC	Landing Craft Air Cushion is also known as Hovercraft or watercraft. These are used for transport and landing of soldier, tanks, cargo from ship to shore. This comes under the collaborative marine forces.
Unit	A unit is an individual ship in a force
Salvo	A salvo is the number of shots fired as a unit of force in a discrete period of time.
Combat Potential	Combat potential is a force's total stored offensive capability of an element or force measured in number of total shots available.
	The capacity for forces to successfully engage in combat is called combat potential. The combat potential of forces can be measured in terms of their designed and available potential. The designed combat potential of forces is the

Term Remarks

capacity of those forces to engage in combat, measured under ideal conditions of training, equipment, organization and motivation. The available combat potential is the potential value of forces in the current state with respect to training, equipment, organization, motivation, geographic and weather environment, and specific enemy (Hughes, 1993).

Combat power

Also called **striking power**, is the **maximum offensive capability** of an element or force per salvo, measured in the number of hitting shots that would be achieved in the absence of degrading factors.







Term Remarks

Figure 22. Visualization of three key concepts i.e., Force, Combat potential, Combat Power (*Hughes*, 1993).

Model

Model is an abstract representation of the physical system. This abstract representation can be mathematical equations. Mostly these are ordinary differential equations. Various types are

Term		Remarks	_
Type of Models	Example	Solving Methods	Limitations
Empirical	P(t)=a+bt+ct ²	Ordinary least square	Less confident about predict outside the range of the data, parameters a,b,c has no meaning
Stochastic	Birth death Process	Probabilistic	Complicated
Simulation	Computer Program, Engineering models	Random Sampling, Simulation	Gives most realistic models but not the best models.
Deterministic	Models and sub model, no randomness in the model	Differential Equations	All relationship model of the components is not known.
Statistical	Hypothesis Testing	Statistical Analysis	Data is not available always

Table 7. Different Types of Models (Barnes & Fulford, 2009)

Term	Remarks
ORBAT	Order of Battle, hierarchical command structure, strength, disposition of personnel, equipment of units and formations of the armed force.
Scouting efficiency	Scouting efficiency is the dimensionless degradation factor applied to a force's combat power as a result of imperfect information. It is a number between zero and one that describes the difference between the shots based on perfect knowledge of enemy composition and position and shots based on existing information.
SAGE algorithm	C2 planning algorithm generally developed in TLC model to support the development of adaptive strategies for policy analysis. This algorithm search for best strategies which minimizes own damage or maximizes opponent's damages. See (Hillestal & Moore, 1996) for further reference.
Simulation	Simulation is an interactive, distributed or standalone, method of learning the properties/behaviour of one or more interacting components (or models) of a complex system. M&S represents building the domain specific decision-making environments from a model-based system perspective, which includes all the components of the system in the form of abstract
	Caratana Danian

Term

Remarks

views of the models. These environments are configured and scaled to examine new tasks, operations, functions, and missions at various levels of system. Random inputs and observing corresponding output are the basic purpose of conducting simulation.

System

A **system** is defined as an **aggregation** or assemblage of **objects** which interact to accomplish a set of goals or **objectives**. [N.K.Jaiswal pp.59]

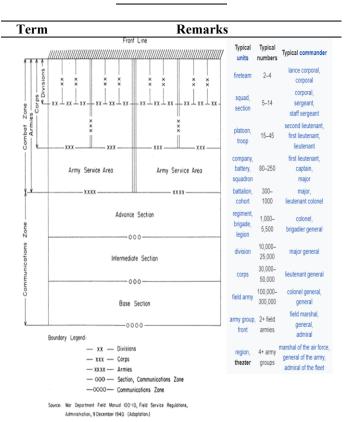
Systems Analysis The basic thought behind Systems Analysis can be explained as "a systematic approach to helping a decision maker choose a course of action by investigating his full problem, searching out objectives and alternatives, and comparing them in the light of their consequences, using an appropriate framework insofar as possible analytic - to bring expert judgment and intuition to bear on the problem" (see (Quade & Boucher, 1968)).

Training Effectiveness

Training effectiveness is a fraction that indicates the degradation in combat power due the lack of training, motivation or readiness. Term Remarks

Theatre

The entire land, sea, and air area that may become or is directly involved in war operations



Term	Remarks			
Theater	Four characteristics (FRMR: Flexibility, Resolution,			
Level	Maneuver, Resource Allocation) of theater-level campaign are			
Campaign	1) FLEXIBILITY: Regular grid versus Irregular structure			
	RESOLUTION: Varying the level of resolution and Cross-resolution modeling 3) MANEUVERABILITY: modeling maneuver at the campaign level, maneuvering forces in coordination with information, space intelligence, faster decision making and situational awareness 4) RESOURCE ALLOCATION: Adaptive Resource allocation: to find the optimal strategies through AI or OR techniques.			

See (Hillestal & Moore, 1996) for further reference

INDEX

I joint operation, 16 Α aggregated combat modeling, 24 M aggregation, 20, 24 Anti-tank, 28 models, 11, 20, 24, 39 attrition rates, 24, 26, 27 0 В ORBAT, 38 Brigade, 22 P C probability, 23, 26, 27 CSRS, 28, 30 S D simulation, 11, 24, 25, 26, 34, deterministic, 25 39 Division, 22 Simulation, 24, 28, 34, 39 strategic, 34 System, 28, 30 G Group, 17, 18, 28, 34 7. zero, 38 Н

heterogeneous, 19, 20, 21, 24

References

Allen, P. (1997). Situational Force Scoring: Accounting for
Combined Arms Effects in Aggregate Combat
Models. Naval Postgraduate School, RAND. USA:
RAND.

B.C. Graham. (June 1979). The COANEW Model. Forty-third

Military Operations Research Society Symposium, .

West Point, New Work..

Barnes, B., & Fulford, G. R. (2009). Mathematical Modeling
With Case Studies: A differential Equations
Approach Using MapleTM and MATLAB@. New
York: CRC Press.

Bourdon, S. (2014, April). A method for evaluating Mission

Risk for the National Fighter Procurement

Evaluation of Options. Retrieved from Government

of Canada: www.canada.ca

Caldwell, B., Hartman , J., Parry, S., & Washburn, A. (2000).

Aggregated Combat Model. US: Naval Postgraduate School.**

D. S. Alberts, J. G. (1999). *Network Centric Warfare,*Developing and Leveraging Information Superiority

(2nd ed.). CCRP publication series.

Dupuy, T. N. (1995.). *Attrition: Forecasting Battle Casualties*and Equipment Losses in Modern War. Washington,
DC. USA: Nova Publications.

Dutta Debasis, S.-C. (Draft). *Methodology to the Evaluation*Fighter Aircraft Alternatives. New Delhi:

Unpublished.

Eric Heginbotham, e. a. (2015). *The U.S.-China military* scorecard: forces, geography, and the evolving balance of power, 1996-2017. Santa Monica, Calif.: RAND Corporation. doi: ISBN 978-0-8330-8219-0.

F. Bolderheij, e. a. (2005). Risk-based Object-Oriented
Approach to Sensor Management. 8th International
Conference on Information Fusion, (pp. 25-29).
Philadelphia (PA).

G.M. Clark. (1969). *The Combat Analysis Model,* . Columbus, Ohio,: Ph.D. Thesis, The Ohio state University, .

Operation Research, 7, 30-44.

H.K. Weiss. (1975). Lanchester-type Models of Warfare.

H. Brackney. (1959). The Dynamics of Military Combat.

Proceedings of the First International Conference on Operational Research, (pp. 82–98).

Hillestad, R. J., Bennett, B., & Moore, L. (1996). *Modeling for campaign analysis.* U.S.: RAND.

Hillestal, R. J., & Moore, L. (1996). The theater level campaign model: A research prototype for a new generation of combat analysis model. Washington D.C.: RAND.

Hughes, J. W. (1993). *Combat Analysis for command, control and communications : A Primer.* Monterey,

California: Naval Postgraduate School.

IDS. (2018). Joint Doctrine of Indian Armed Forces. New Delhi.

IEEE:1220-2005. (2005). *IEEE standard for Application and Management of the system engineering Process.*Institute of Electrical and Electronics Engineers.

IEEE:15288.2-2014. (2014). IEEE standard for Technical
Reviews and Audits of Defense Programs. IEEE.

ISO/IEC/IEEE:15288. (2015). System and Software

Engineering-System Life Cycle Processes. IEEE.

ISO/IEC/IEEE:15289:2015. (2015). Content of Systems and Software Life Cycle Information Products. IEEE.

ISO/IEC-TR24748-1:2010. (2010). Guide for Life Cycle Management. IEEE.

ISO/IEC-TR24748-2:2011. (2011). *Guide for Application of 15288.* IEEE.

ISO/IEC-TR24748-3:2011. (2011). *Guide for Application of 12207.* IEEE.

ISO-10303-AP233. (n.d.). Industrial Automation Systems and
Integration-Part 233: Systems engineering data
representation. ISO.

J. Andrighetti. (September, 1973.). A Model for the Statistical
Analysis of Land Combat Simulation and Field
Experimentation Data, . Monterey, California,: M.S.
Thesis in Operations Research, Naval Postgraduate school, .

J.G. Taylor. (1983). Lanchester Models of Warfare (Vol. I &
 II). Alexandria, VA: Military Applications Section,
 Operations Research Society of America,
 Alexandria, VA. 1983.

J.G. Taylor, U. Y. (2000). Hierarchy-of- models Approach for Aggregated-force Attrition. *Proceedings of 2000 Winter Simulation Conference*, (p. 925). USA.

Jaiswal, N. K. (1997). *Military Operations Research: A*quantittaive Decision Making. New York: Springer
Science+Business Media, LLC.

(July, 2013). *Joint Operation Planning Process (JOPP)*Workbook. JMO Department, Naval War College.

Lanchester, F. (1914). Aircraft in Warfare: The Dawn of the Fourth Arm No. V., The Principle of Concentration", , pp. 422–423, 1914. *Engineering*, 422-423.

(2004). Naval System Engineering Guide.

Parry , S. H. (1992). Evaluation of Attrition Methodologies for Combat Models. Naval Postgraduate School Notes. Quade, E. S., & Boucher, W. I. (1968). *Systems Analysis and Policy Planning: Applications*. New York: American Elsevier.

R. G. Stockton. (1973). CARMONETTE-Division Battle Model
Interface. Proceedings of the Twelfth Annual U.S.
Army Operation Research Symposium, (pp. 23-32).

R.L. Helmbold. (1965). A Modification of Lanchester's Equations. *Operation Research*, *13*, 857-859.

Ramazan, G. (2000). Fitting firepower score models to the battle of Kursk data. USA: Naval Postgraduate School.

Stromberg, D., & et., a. (2002). On Platform-Based Sensor

Management. Fifth International Conference on

Information Fusion.

Technium, [. T. (2020, July 16). *The Technium*. Retrieved from The speed of Information,: https://kk.org

US Army's Center for Strategy and Force Evaluation.

(September 1998). Kursk Operation Simulation And Validation Exercise - Phasell (KOSAVE II) The US Army's Center for Strategy and Force Evaluation Study Report CAA-SR-98-7. US Army's Center for Strategy and Force Evaluation, US ARMY CONCEPTS ANALYSIS AGENCY. MARYLAND: OFFICE OF THE CHIEF OF STAFF.

V.K. Rohatgi. (1984). *Statistical Inference*. USA: Wiley series in probability and mathematical statistics.