Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

APPLYING DATA FARMING FOR MILITARY OPERATION PLANNING IN NATO MSG-124 USING THE INTEROPERATION OF TWO SIMULATIONS OF DIFFERENT RESOLUTION

Daniel Huber

Fraunhofer Institute for Intelligent Analysis and
Information Systems
Schloss Birlinghoven
53757 Sankt Augustin, GERMANY

Daniel Kallfass

Airbus Defence and Space GmbH

Claude-Dornier-Straße
88090 Immenstaad, GERMANY

ABSTRACT

The NATO Modeling and Simulation Group (MSG) set up the task group MSG-124 in 2013 to provide NATO with actionable decision support using the data farming methodology. This paper presents a large-scale data farming experiment conducted within MSG-124. Two connected battlefield simulation tools – PAXSEM and ITSimBw – are used to get insight into possible courses of actions for the decision-maker in NATO Operation Planning. In a conventional warfare scenario PAXSEM is used to simulate an airstrike and entry phase on single-entity level. For the subsequent simulation of a massive land attack phase on an aggregated level, the simulation state is transferred to ITSimBw using the Military Scenario Definition Language (MSDL). This paper covers the scenario and model definition, data farming experiment set-up, and input and output data processing necessary to conduct this large-scale simulation with interconnected models of different resolution levels.

1 INTRODUCTION

In 2010, the NATO Science and Technology Organization (STO) started the three-year Modeling and Simulation Task Group MSG-088 “Data Farming in Support of NATO” to assess and document the potential of the data farming methodology in use for decision support (Horne et al. 2013). Its successor, the MSG-124 Task Group “Developing Actionable Data Farming Decision Support for NATO”, was initiated to transform the data farming methodology described in MSG-088 into an actionable form, i.e. to give actionable support for NATO decision-makers. The MSG-124 started its work in 2013 and is scheduled to complete at the end of 2016. Participating nations are USA, Germany, Sweden, Finland, Canada, Australia, Turkey, Norway and Italy. It is split into two syndicates: “Cyber Defence” (not covered in this paper) and “Operation Planning” – each syndicate perusing the objective of MSG-124 in different application areas. The “Operation Planning” syndicate of MSG-124 aims at developing a decision support tool for utilizing data farming in support of course of action development, analysis and comparison, exemplarily adapted to the activities of the Joint Operation Planning Group on Joint Forces Command level according to the NATO Comprehensive Operations Planning Directive (COPD, Version 2.0, NATO 2013).

The term “Data Farming” was introduced by Brandstein and Horne (Brandstein and Horne 1998). It is a process that has been developed in order to support decision-makers in answering questions that are not addressed by traditional modeling and simulation processes (Horne and Meyer 2004). The data farming process consists of six realms: “Collaboration”, “Model Development”, “Rapid Scenario Prototyping”, “Design of Experiments (DoE)”, “High Performance Computing” and “Analysis and Visualization” (Horne and Meyer 2010). The core idea of data farming is the conduction of a very large number of simulation runs on high-performance supercomputers. Numerous input parameters are varied along a deliberately defined plan, the outputs are measured and finally the mutual interrelation are examined (Kallfass and Schlaak 2012). To demonstrate the benefit of data farming for actionable decision support in operation planning a large symmetrical warfare scenario was chosen. The details of the scenario are presented in detail in the next section. The first two phases of the scenario have a limited number of participating entities. For modeling and simulation of these phases, the single entity simulation tool PAXSEM developed by Airbus Defence and Space was chosen. The third phase has more than 5000 participating single entities and cannot be efficiently modeled and simulated on single entity basis. Thus, the planning and simulation tool ITSimBw developed by Fraunhofer IAIS was chosen to model and simulate the third phase on an aggregated battalion level. Using two different simulation tools and models of different resolution requires the elaborate coupling of both tools to generate a coherent simulation state and output data. This paper presents how the coupling is realized using standardized interfaces and how this approach is integrated in a large-scale data farming experiment.

2 THE BOGALAND SCENARIO

The Scenario chosen was provided by the Swedish Defence Research Agency (FOI) and is a derivative of the “Bogaland Scenario”, that depicts a large-scale conventional military operation. In the scenario, the operation is separated into 3 different phases: airstrike, entry and land attack phase. The airstrike and entry phases consists mainly of air-to-air and air-to-ground engagements. The subsequent land attack phase consists of battalion or brigade level engagements with fighter support.

2.1 Scenario Overview

A virtual country, Bogaland, geographically located in the southeast of Sweden, is threatened by its northern neighbor Northland (Figure 1). Northland intends to occupy parts of Bogaland in four phases and to deny the Blue Forces (BFOR, i.e. NATO allied forces) involvement in this conflict.

Bogaland defense plans are based on the support of coalition forces. Therefore, the operational objective of Bogaland’s armed forces in the first phase of a Northland attack is to delay Northland’s advance in order to gain time for BFOR deployment.

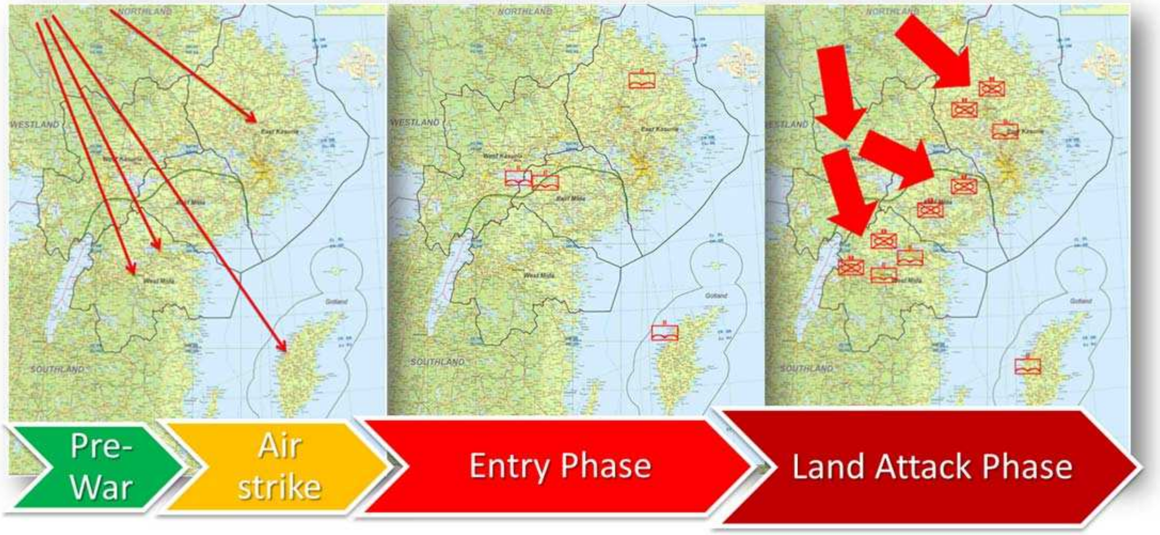

Figure 1: Bogaland scenario overview.

After a pre-war activity, the offensive starts with an airstrike phase (see Figure 1 left). Northland air forces conduct primarily an offensive counter-air operation in gain air superiority. The Northland air attack forces use cruise missiles, multi-role fighters and bombers against Bogaland's aircraft, air defense installations and airbase infrastructure.

Bogaland's air force deploys multi-role fighters in combat air patrols or as intercept fighters. Air defense units are deployed in order to provide area defense to Bogaland's 4 airbases. In addition, after a short response delay, BFOR will deploy fighter aircraft to reinforce Bogaland's air force defensive air operations.

The entry phase of Northland's airborne forces starts when the air defense units on Bogaland's airbases are destroyed and the runways are still operable. Airborne infantry forces are deployed using transport planes during the entry phase to seize Bogaland's 4 airbases (Figure 1 center) to deny their use by BFOR and to establish bridgeheads in support of the subsequent land attack phase.

After the entry phase, the land attack phase is initiated by Northland (Figure 1 right). Land units consisting of up to 40 armored, mechanized infantry and artillery battalions attack their operational target areas and are supported by the air forces that survived the airstrike phase. Northland can choose between several different strategies for their ground assault, varying the strength of the attacking force for each target area. Bogaland also has several possible strategies for how to deploy their defense forces (up to 15 land battalions).

2.2 Main Study Question

The main question to be answered by this study is (colloquial form): "*How can we – Bogaland – best use what we have - including NATO Response Force?*"

The goal is to provide decision support for building one or more courses of action that enable Bogaland to robustly resist attacks from Northland under given constraints. Therefore, the commander's priority is to hold important infrastructure and areas of Bogaland, delay the beginning of the land phase and set good conditions for the subsequent deployment of coalition forces (not included in this scenario). It shall be investigated which unit types, quantities, equipment and tactics, techniques, and procedures (TTPs) in each phase are most robust against Northland's most likely and most dangerous courses of action.

3 SCENARIO REALIZATION WITH PAXSEM AND ITSIMBW

3.1 Introduction of PAXSEM

The agent-based simulation framework PAXSEM was developed by Airbus Defence and Space on behalf of the German Armed Forces since 2008. Its main areas of application are simulation-based military analyses in Concept, Development and Experimentation (CD&E) experiments and the support to foreign missions as well as to national procurement activities.

Its high resolution 3D visualization allows the simulation of technical-tactical scenarios and plots. Within these, military units are represented as agents in a granularity ranging from single entity to enforced company level. PAXSEM enables a detailed, physically based representation of technical systems equipped with sensors (e.g., optical/infrared/radar) and effectors (e.g., small armed fire/artillery/rockets/guided missiles). These agents independently execute activities, perceive their environment (sensors), intervene therein (effectors) and react upon the changes in the respective ambience (dynamics). This allows multiple differential decision alternatives to be created, describing a wide sample space. The mutual interference of individual agents allows simulation of complex systems that are often impossible to predict and uniquely confined by the agents abilities (Kallfass and Schlaak 2012). Hence, valuable insights can be generated regarding the operational employment e.g., defined tactics, techniques and procedures (TTPs) of systems in use, systems under development or systems that do not yet exist at all.

3.2 Introduction to ITSimBw

ITSimBw is a long-term project of the German Federal Armed Forces and the Fraunhofer Institute for Intelligent Analysis and Information Systems. It began as a multi-agent simulation system (Hügelmeyer et al. 2006, Hügelmeyer et al. 2007), but over time focus shifted to planning and macro-simulation of military applications and operations. In several studies, interfaces to PAXSEM using Battle Management Language (BML) and Military Scenario Definition Language (MSDL) for interoperability on multiple resolution levels have been developed and refined. Alongside MSG-088 (Horne et al. 2013), a data farming framework for the Federal Armed Forces and a planning and macro-simulation method of ITSimBw was developed. For MSG-124 the planning and macro-simulation method was refined and extended to cope with more complex scenarios, lower levels of resolution and hierarchical command and control.

3.3 Assignment of Operation Phases to Simulation Models

PAXSEM is well suited to simulate the airstrike and entry phase, since the details of flight movement and guided missile behavior play an important role in combat and simulation accuracy is crucial. The number of air combat and air defense units depicted is no more than a few hundred, which results in a manageable simulation runtime.

As the land phase consists of up to 55 land battalions with more than 100 entities per battalion (≫5000 in total), the land attack phase has to be simulated on an aggregated level. This number of entities would exceed the manageable complexity of the current version of the PAXSEM model. In ITSimBw it is possible to plan and simulate military operations with hundreds of battalions. It uses aggregated attrition models to calculate unit interaction as well as simplified movement and tactics. Thus simulation runtimes are achievable that are fractions of those of PAXSEM in this scenario. The realization of the scenario in PAXSEM and ITSimBw will be introduced in the next two sections.

3.4 Realization of the Airstrike and Entry Phase in PAXSEM

For a comprehensive representation of its simulated environment, PAXSEM simulates all scenarios on a 3D terrain. For the Bogaland scenario, a 400 x 800 km terrain database is generated containing the entire Bogaland country as well as the southern part of Northland, where the airstrike and entry phase is initiated (see Figure 2 left). The 3D terrain is mainly used for movement planning, intervisibility between entities and for weapon trajectory and impact computation.

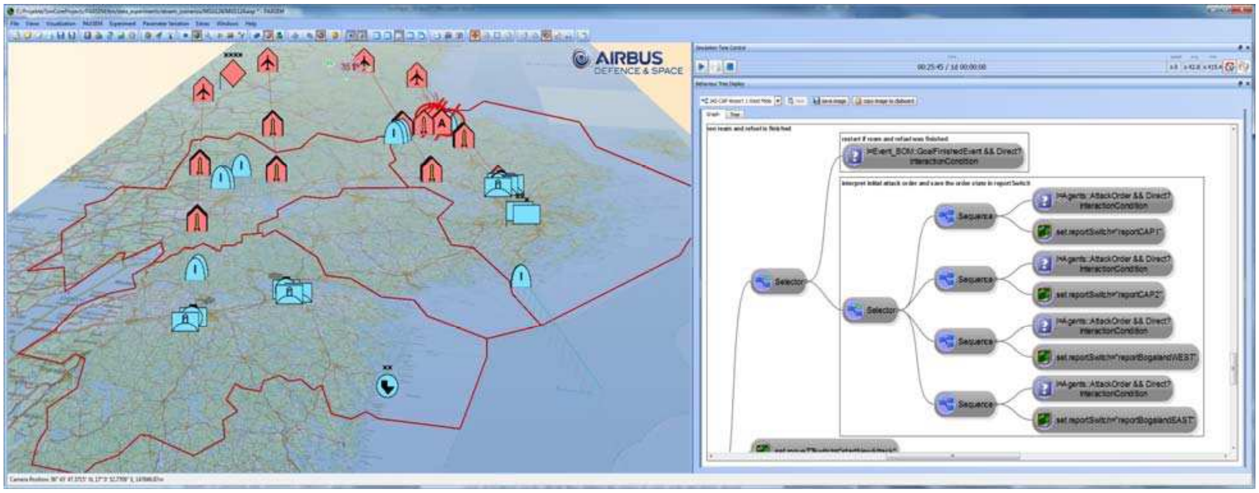
AIRBUS
DEFENCE & SPACE
00:25:45 / 1d 00:00:00
restart if roam and refuel was finished
!=Event_BOM::GoalFinishedEvent && Direct? InteractionCondition
interpret initial attack order and save the order state in reportSwitch
Selector
Selector
Sequence
Sequence
Sequence
Sequence
!=Agents::AttackOrder && Direct? InteractionCondition
set reportSwitch="reportCAP1"
!=Agents::AttackOrder && Direct? InteractionCondition
set reportSwitch="reportCAP2"
!=Agents::AttackOrder && Direct? InteractionCondition
set reportSwitch="reportBogalandWEST"
!=Agents::AttackOrder && Direct? InteractionCondition
set reportSwitch="reportBogalandEAST"

Figure 2: Airstrike phase in 3D and behavior tree example in PAXSEM.

All entities (e.g., the JAS Gripen or F-18) are configured in PAXSEM with their tactical symbols and 3D models as well as their technical parameters, such as maximum speed or flight level in specific mission situations. In addition, sensor and weapon systems of all entities are configured to include their hit and kill probabilities. Most of the required weapon data was provided by the Swedish Defence Research Agency (FOI), in a sanitized, not restricted, quality.

The general behavior of all agents is flexibly definable through an IF-THEN-rulesets involving a behavior tree methodology (see Figure 2 right). The rulesets are put into graphs supporting the definition monitoring and tracking of the agents' behavior. In addition to basic behavior actions, such as move or attack, a behavior tree usually consists of additional control nodes like sequences, selections, conditions, loops, parallel tasks etc. An arbitrary set of behavioral rules can be implemented for any individual entity, which are activated once a relevant condition delivers the trigger thereto. Triggers can be specific sensor information, presence of entities in crucial areas, weapon threats, ammunition engagements, or other.

By this way of modeling entity behavior, the JAS Gripen fighter, for example, has the behavior to start and engage hostile air targets based on a predefined target prioritization and has to land on its airbase as soon as it runs out of ammunition or fuel. In case its airbase is destroyed, the fighter has to land on the closest alternative airbase. The transport of the airborne battalions was modelled by dividing each battalion into 20 platoons which can be transported by one transport plane each. Depending on the number of airborne and transport planes, several delivery runs are made, each with fighter escort.

3.5 Realization of the Land Phase in ITSimBw

The simulation environment of ITSimBw consists of two separate modules – planner and simulation – that interact using BML and MSDL. By using these standardized interfaces it is possible to connect the planning module to other simulation models, e.g., models of different scope, level of abstraction, etc.

The planner is based on the methodology of Capability Based Planning and uses a hierarchical mind-map-like modeling methodology. This model consists of the used capabilities, available resources, resource hierarchy and geographical locations.

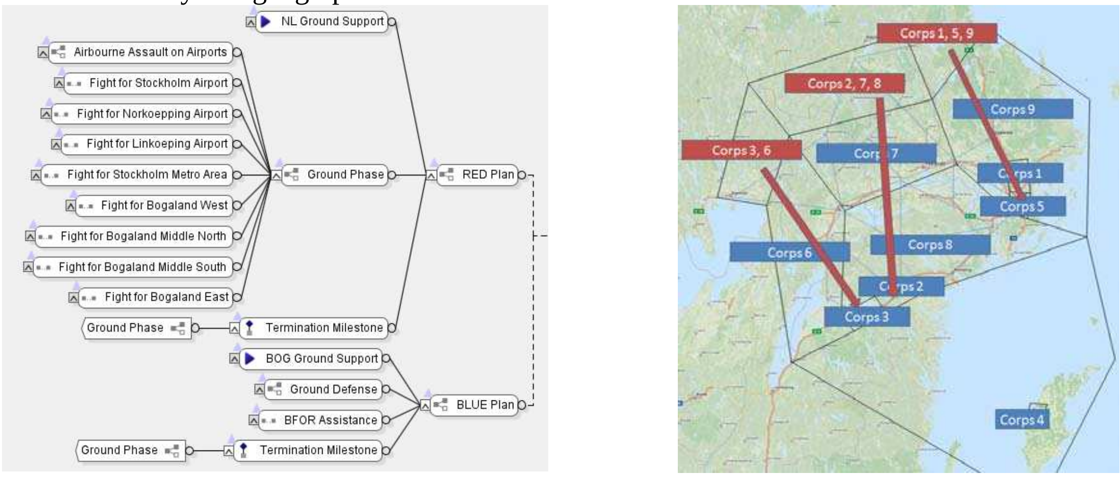

Figure 3: Operation chart of the plan and plan visualization.

The main part of the model is the actual operation plans (Figure 3 left). These define how two or more parties use capabilities to disable opposing resources and thus remove capabilities from the opponents portfolio. Modelled capabilities include "Air Support", "Defend Area" or "Attack Opposing Resources in Area", among others. Each capability needs certain types of resources to be executable. The plan then defines which specific unit resources of that type are assigned to execute the capability.

The plans for the Bogaland scenario are shown in Figure 3 left, the "RED Plan" defining the courses of action of Northland and the "BLUE Plan" defining the courses of action of the Bogaland coalition.

Several different kinds of elements are used in this plan; the three most important elements being step (triangle, actual capability use), sequence (two squares, sequential execution of sub-elements) and parallel execution (three squares, parallel execution of sub-elements). The plans are analyzed iteratively over simulation time by the planner. In each iteration all steps are evaluated if they can be executed, i.e. if the required resources are available (not destroyed and not used in another step at the same time) and if the objective of the step is still unachieved. After each plan evaluation iteration, the steps for execution are transmitted to the macro simulation in form of BML orders. The simulation then calculates resource movement using an abstract road network, point-shaped resources and resource attrition using an adapted Correlation of Forces (Zanella 2012) model. Simulation of resources continues until one or more orders are fulfilled (objective achieved) or failed (objective can't be achieved anymore). At this point the state of the resources in the simulation are transmitted back to the planner, which then starts a new planning iteration. Prior to the initial simulation start, the resource state has to be transmitted to the simulation from the planner via MSDL.

An aggregated visualization of the plan is given in Figure 3 right. Bogaland is partitioned into nine areas and Northland is abstractly modeled as three areas. The forces of both Bogaland and Northland are organized into corps, each corps consisting of a variable number of either armored, mechanized infantry or artillery battalions. These battalions are dynamically generated prior to the planning execution according to the design of experiment of the data farming experiment. Each corps of Bogaland is tasked to defend one area, and each corps of Northland assigned to attack one area of Bogaland. Depending on the unit disposition in the DoE and thus the assignment of units to corps, the actual simulation of the plan is greatly variable.

4 METHODS OF INTEROPERATION

To allow the continuation of the scenario in the land phase, the simulation state of PAXSEM has to be transferred to ITSimBw. The simulation state is defined by the model entities, e.g. position and status of a fighter at the time of transfer. Since PAXSEM is an entity-based simulation its state is composed of the state of its entities. ITSimBw simulates on an aggregated level (i.e. battalion and squadron), which requires that PAXSEM entities have to be mapped to units in ITSimBw. Most units or entities, however, are only used in either the air and entry phase or the land phase. These units don't have to be mapped and can be created in both tools using their definition in the DoE. This applies to air defense units (Patriots), radars, bombers, transport planes, cruise missiles and armored, mechanized and artillery battalions. Entities or units that need to be mapped are fighters and airborne battalions.

After the land phase is finished, another interaction process is needed to create consistent output data collection. These two processes – mapping the simulation state and collecting output data – are described in the next sections.

4.1 Mapping of the Simulation State

The simulation state consists of the state of the entities and units and the simulation time. To transfer the state, the Military Scenario Definition Language (MSDL) (SISO-STD-007-2008) is used. PAXSEM writes the state of all entities into a MSDL file and saves it in the file system. The location of this file is defined in the DoE-file, a common XML document, which both PAXSEM and ITSimBw use to create their entities and units. Upon start, ITSimBw opens the MSDL and analyses it.

PAXSEM uses the MSDL object *EquipmentItem* to store its entities and *ScenarioTime* to store the time. The time can easily be parsed by ITSimBw, since it has to be conform to ISO 8601. To identify the relevant fighters and airborne, the *SymbolIdentifier* of all equipment items is used. If the symbol identifier contains the sub-string "MFA" (fixed wing attack) and "MFFI" (fixed wing interceptor) according to the Common Warfighting Symbology (MIL-STD-2525C), it is a fighter. The list of fighters is then split into blue and red fighters. To do this split, the *ForceOwnerHandle* of the equipment items is analyzed by comparing it to the *ObjectHandle* of the global object *ForceSides*. The type of fighters is not distinct in

ITSimBw, only the amount is relevant. The fighters for both sides are organized into one squadron each and placed on one randomly chosen operational airport. Since the time advance model in ITSimBw is much coarser and the area of operation is rather small in relation to the fighter speed, the position of the fighters is disregarded and they are considered landed and ready for operation.

The airborne battalions are transported in transport plane-sized portions, i.e. 1/20 of a battalion, called a platoon. As a symbol identifier, these platoons have a sub-string “EWR”, i.e. a weapon according to the Common Warfighting Symbology. A list with all platoons is created. Since the success of transportation can only be identified by evaluating the position of the platoon, the set of all platoons is split into 4 set of platoons close to the 4 airports and the rest. The split is performed by comparing the *Location* of the platoon with the position of each airport. If the distance is less than 5 km, it is considered successfully transported. All other platoons was either never transported or the transport plane was shot down during flight. For each airport, airborne battalions are created by defining 20 platoons as a fully operational airborne battalion. If there are less than 20 platoons or there is a rest, a partly operational (in percent of fully operational) battalion is created. The position of the battalion is set as the position of the airport.

4.2 Collecting the Model Outputs

All model outputs of PAXSEM are written into a comma separated values (CSV) file, starting with a header line, describing all simulation output columns, and continued with one line of model outputs, describing the simulation state at a predefined simulation time or at an event or at the end state of the simulation run. Since there needs to be one coherent output file for the whole scenario, ITSimBw writes its simulation outputs to the same file. Therefore, the CSV filename is passed to ITSimBw when started after PAXSEM to ensure that identical files are used. ITSimBw extends the header line in the CSV file with column descriptions of its additional simulation output and appends its simulation output values to the last line of the CSV file, which already contains the simulation output of the simulation end state of PAXSEM.

In addition, ITSimBw can be executed with an option that executes several simulation runs with varying input factors only relevant to the land phase (e.g., strategy combinations of red and blue). Therefore, the simulation output data of PAXSEM in the last line of the CSV file needs to be duplicated and appended by the simulation output of each ITSimBw simulation run. This approach ensures that the data farming environment can merge the results of all data farming simulation results into one single CSV file such that it can be easily analyzed by using statistical tools like e.g., SAS JMP.

5 THE DATA FARMING EXPERIMENT

5.1 Introduction to the Data Farming Process

As already been mentioned in Section 1, the data farming methodology is designed to answer question and gain insights by combining the six realms of data farming:

1. Collaboration: Working in collaborative teams with subject matter experts, model experts, analysts, etc.
2. Model Development: Building and extending data farmable simulation models.
3. Rapid Scenario Prototyping: Depicting the scenario in the simulation model and discussing it with subject matter experts, such that the scenario can be used to answer the initial questions.
4. Design of Experiments (DoE): Defining the variation of the input factors in specific value ranges intelligently to reduce the combinatory complexity of all factor combination possibilities.
5. High Performance Computing: Computing all required simulation runs on a high performance PC cluster.
6. Analysis and Visualization: Statistically analyzing and visualizing the “Measures of Effectiveness” (MoEs) of all simulation runs.

The high number of simulation runs and wide input parameter ranges allow robust insights to be obtained; finding the most important factors that influence the outcome, analyzing outliers and answering "What if?" questions.

The subsequent sections describe the design of experiment, the high performance computing, the measures of effectiveness and the analysis and visualization of the Bogaland scenario.

5.2 Design of Experiment (DoE)

According to the scenario, the following decision factors (definition of the blue forces) and noise factors (definition of the red forces) are defined in the following two tables:

Decision Factor name	Type	Range	Description
# F18	Integer	12-24	Number of F18 of BFOR
# JAS	Integer	12-36	Number of JAS of Bogaland
# JAS / CAP	Integer	2-6	Number of JAS per CAP (Combat Air Patrol)
# CAP	Integer	0-2	Number of CAP
# Patriot	Integer	8-12	Number of Patriot
# Battalions	Integer	10-15	Number of Battalion
NATO Arrival Time	Hours	2-10	Arrival time of BFOR F18 after operation start
Weapon Mix	Categorical	3	Short, medium or long range JAS weapons
Patriot Disposition	Categorical	3	One, two or no cluster of Patriots
Strategy	Categorical	4	Defense focus on areas : front, east, airports, all

Noise Factor name	Type	Range	Description
# Attack Packages	Integer	6-10	Number of attack packages (each containing 8 fighters + 2 bomber)
# Airborne	Integer	4-8	Number of airborne battalions
# Transport Plane	Integer	20-40	Number of transport planes
# Cruise Missile	Integer	24-48	Number of cruise missiles
# Battalions	Integer	30-40	Number of battalions
CM burst size	Integer	2-3	Number of cruise missiles per burst
CM burst delay	Minutes	15-30	Time between bursts
Strategy	Categorical	4	Attack focus on areas: front, east, airports, all

Since it is very beneficial to analyze the quality of courses of action for Blue as well as for Red, the decision and noise factors are crossed to ensure that each decision factor combination of Blue runs against each noise factor of Red. Therefore a NONB design (Nearly Orthogonal, Nearly Balanced) (Vieira 2012) with 512 design points for the decision factors and a NOLH design (Nearly Orthogonal Latin Hypercube) (Sanchez 2011, Sanchez et al. 2014) with 32 design points for the noise factors are chosen. The categorical variables "Weapon Mix" and "Patriot Disposition" are integrated into the NONB design, as the NONB allows to handle categorical factors as well. The categorical "Strategy" factors for Blue and Red are not included in the NONB/NOLH-designs but crossed and therefore varied individually. The crossing of the strategies is chosen primarily to allow better analysis and additionally because of software architectural reasons. Since the simulation runtime of the air and entry phase is much higher than that of the land phase, crossing the strategies can be implemented by simply running the land phase 16 times per design point of the air and entry phase and thus not increasing the overall runtime significantly. The resulting number of design points (DP) of the whole DoE is:

$$DP = 512 \times 33 \times 4 \times 4 \ = \ 270{,}336$$

To keep the computation requirement in bounds, the number of simulation replications (simulations with different random seeds) is set to 5. Thus, the final number of simulation runs is 1.35 million runs. An additional step taken to allow better analysis is using specific step sizes for each factor, e.g., the number of battalions is varied in increments of 5. The NONB design offers the possibility to set step sizes, but the NOLH does not. To create step sizes in the NOLH, the original design was done using the number of steps s_i and afterwards calculating the factor values v_i by using a linear function, $v_i = a + bs_i;\ s_i = 0,1,2$ .... By using linear functions, the properties of the NOLH design are not affected.

5.3 Measures of Effectiveness (MoE)

The development of the MoEs is driven by the main study question and the commander's intent presented in Section 2.2. The weighted commander's intent is assumed to be the following:

1. Hold Stockholm area
2. Hold as many areas as possible
3. Delay start of land phase
4. Generate favorable conditions for future operations
5. Keep Airports under control and with active Patriot systems

Favorable conditions are assumed to be achieved by minimizing blue losses and maximizing red losses. The following table shows the selected MOEs.

MoE Name	Type	Range	Symbol	Description
Owner Stockholm	Binary	0,1	O_s	Owner of Stockholm: 0=Red; 1=Blue
# blue Areas	Integer	[0..9]	A_B	Number of areas under Blue control
End-time entry phase	Time	>= 0	T_{EE}	Time at the end of the entry phase
Red and Blue Losses	Double	[0..1]	L_{RA},L_{RB}, L_{BA}, L_{BB}	Relative losses airplanes and battalions on both sides
# blue Airports with active Patriot	Integer	[0..4]	A_{BAP}	Number of airports under Blue control and active Patriot system

For a multi-criteria analysis of the overall mission success the following objective function is defined:

$$OS = w_1\, O_S + w_2 \frac{A_B}{9} + w_3 \widetilde{T_{EE}} - w_4 L_{BA} - w_5 L_{BB} + w_6 L_{RA} + w_7 L_{RB} + w_8 \frac{A_{BAP}}{4}$$

where $w_{1..8}$ are weights and $\widetilde{T_{EE}}$ is the normalized end time of the entry phase in the interval [0..1]. The weights are chosen as: w_1=10, w_2=8, w_3=w_4=w_5=6, w_6=w_7=4 and w_8=2 to implement the priorities in the commander's intent. With these weights, the value range of *OS* is [-12, 26]. For a better understanding when analyzing, *OS* is normalized to $\widetilde{OS}$. For optimizing mission success the objective function has to be maximized.

5.4 High Performance Computing

The German Armed Forces own several high performance computing (HPC) cluster systems. One cluster system containing 512 computer nodes which was used for this data farming experiment is hosted by Airbus Defence and Space in Unterschleißheim, Germany. A data farming software infrastructure with a web application to manage, monitor and conduct the data farming experiments on a the PC cluster is available. The open-source software "HTCondor", developed by the University of Wisconsin–Madison, is used here as a job scheduler to distribute all simulation runs to the available computer nodes. After all the data farming simulation runs are conducted, the simulation output results are merged to a CSV file which can be downloaded through the web application. In addition, a software tool called "Data Farming GUI" is used to define the data farming input parameter variations and design of experiment in an easy and simulation model-independent way.

6 RESULTS

The data farming experiment results in a simulation output table with 1.35 million rows, each row with the output of one simulation run. The execution of these runs on a cluster with 512 nodes took about 75 hours. On average, for each design point, PAXSEM required 15-20 minutes and ITSimBw required 4 minutes (including 16 different land attack strategy combinations for red and blue). The percentage of non-terminating or failed runs in the data set is lower than 0.01% and is therefore ignored. The 1.4 GB sized CSV-output file is analyzed using SAS JMP 11.

Table 1: Correlation between Blue ownership of areas and number of airborne landed there.

		#Airborne landed at			
		Stockholm	Norköpping	Linköping	Visby
Blue Ownership	Stockholm	-0.28	0.09	0.12	0.05
	Norköpping	0.15	-0.45	0.18	0.12
	Linköping	0.13	0.12	-0.32	0.08
	Visby	0.22	0.23	0.23	-0.86

Table 1 shows the correlation of the number of landed airborne battalions at an airport and the final occupant of the airport at the end of the land phase. The MoE "#Airborne landed at" is taken from MoEs generated by PAXSEM, "Blue Ownership" is a MoE of ITSimBw. A significant change in correlation of areas with and without landed airborne units shows the significant effect of PAXSEM simulation results on ITSimBw results. This indicates a successful coupling of both tools.

In Figure 4, four more example output analysis are presented. The contour plot in the upper left shows the average normalized overall success in relation to the number of fighters and battalions. It can be seen that fighters are important for a positive outcome for Blue and that a shortage in the number of battalions can be compensated for by more fighters. The bar chart in the lower left shows the significant influence of the number of battalions, F-18 and JAS Gripen on average normalized overall success given a specific red COA. Given the same red COA, the bar chart in the upper right shows that the blue strategy "East", which is protecting the area around Stockholm and its airport is clearly favorable in comparison to the others. Finally, the bar chart in the lower right shows the significant influence of the armament of the JAS Gripen in relation to the average red airplane losses, favoring long range missiles.

7 CONCLUSION AND FUTURE WORK

This paper demonstrates that the coupling of existing simulation models of different resolution is reasonable and beneficial for simulating large military operations. Depending on the question base and scenario needs, the best suitable simulation resolution can be selected for each scenario phase. The interoperation using the SISO standard MSDL has proven to be a valuable support in the exchange of simulation states in military environments. Through its standardization, MSDL allows the quick replacement of simulation tools and models that are MSDL compliant. In addition, this paper has shown that data farming experiments can be conducted in a setup of coupled simulation models, as long as some frame conditions are considered (e.g. taking into account the individual computation time of each simulation model when setting up a common DoE).

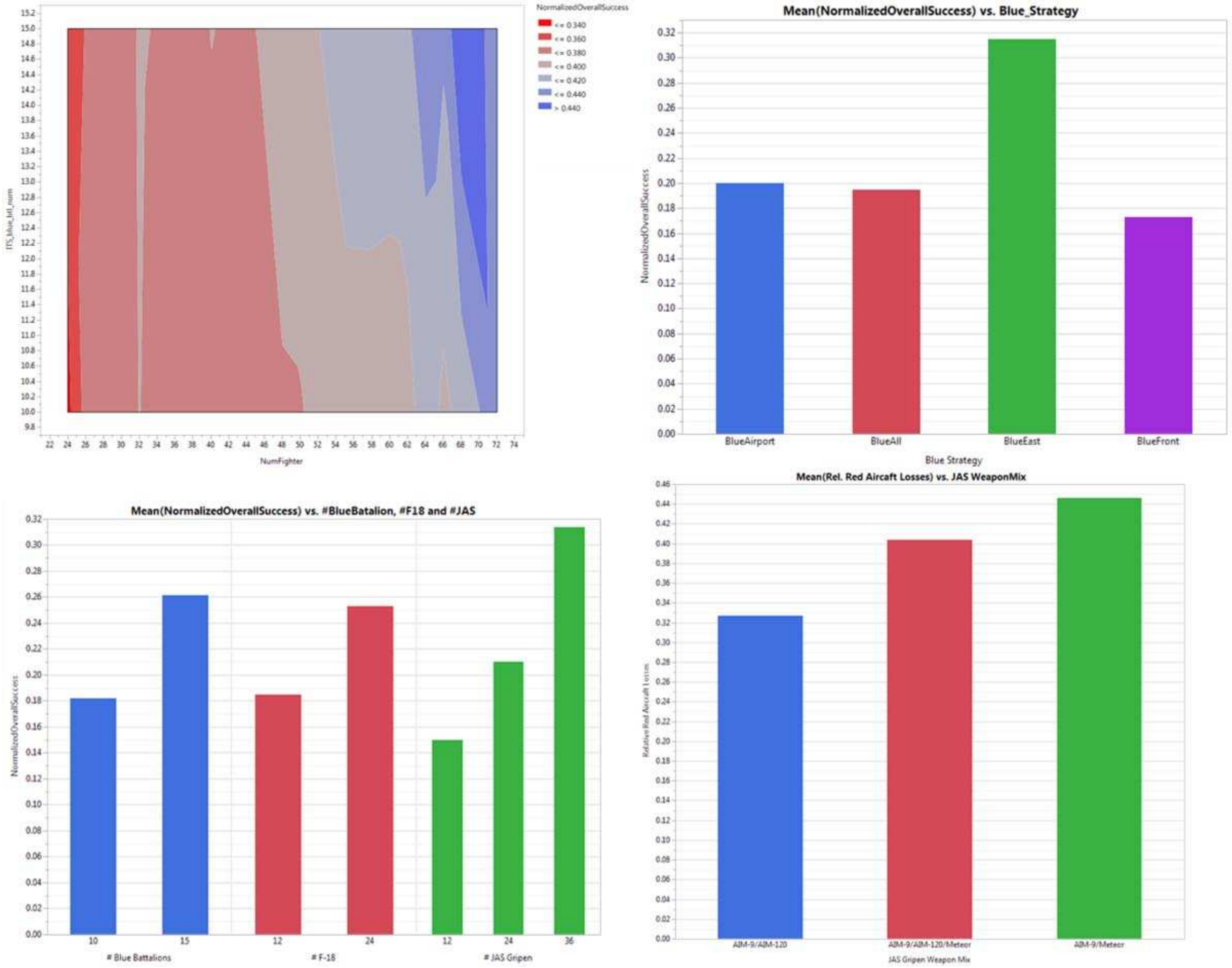

Figure 4: Data farming results.

As future work, the NATO MSG-124 “Operation Planning” syndicate plans to extend the coupling of PAXSEM and ITSimBw with the objective of simulating specific combat situations of the land phase in detail in PAXSEM. For such hotspot analysis, a part of the scenario situation has to be transferred via MSDL and BML, disaggregated and simulated with PAXSEM on single entity level. The simulation results finally need to be aggregated and fed back to ITSimBw. In this specific setup with PAXSEM and ITSimBw, the outlined approach will bring along an augmented accuracy of the overall results. The increase in accuracy is not only achieved by the increased simulation accuracy at hotspots but its outputs can also be used as “training data” for the used attrition models of ITSimBw.

ACKNOWLEDGMENTS

The authors would like to express their gratefulness to all members of the NATO MSG-124 task group for their input and support on the syndicate on “Operation Planning”. Thanks also to the MSG-124 co-chairmen Dr. Gary Horne (USA) and LTC Stephan Seichter (Germany) as well as the syndicate co-leads Sebastian Döring (Germany) and Dr. Johan Schubert (Sweden) for allowing us to publish the intermediate results of the syndicate.

REFERENCES

Brandstein, A. and G. Horne 1998. "Data Farming: A metatechnique for research in the 21st century", In *Maneuver Warfare Science 1998*, Marine Corps Combat Development Command Publication, Quantico, Virginia.

Horne, G. and T. Meyer 2004. "Data Farming: Discovering Surprise", In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. Ingalls, M.D. Rossetti, J. S. Smith, and B. A. Peters, 171-180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Horne, G. and T. Meyer 2010. Data Farming and Defense Applications, *Proceedings of the MODSIM World 2010 Conference, Hampton Roads, Virginia*.

Horne, G. et al. 2013. *MSG-088 Data Farming in Support of NATO*. Final Report. NATO Science and Technology Organization (STO), Modeling and Simulation Coordination Office, Paris, France.

Hügelmeyer, P., T. Steffens and T. Zöller. 2006. "Specifying and Simulating Modern Warfare Scenarios with ITSimBw." In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1273-1279. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kallfass, D. and T. Schlaak 2012. "NATO MSG-088 Case Study Results to Demonstrate the Benefit of using Data Farming for Military Decision Support." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 2481-2492. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sanchez, S. M.. 2011. "NOLHdesigns spreadsheet." http://harvest.nps.edu/ [Accessed March 31, 2015]

Sanchez, S. M., P. J. Sanchez and H. Wan 2014. "Simulation Experiments: Better Insight by Design". In *Proceedings of the 2014 Summer Simulation Multiconference*, SCS ACM Publishing.

SISO-STD-007-2008. 2008. *Standard for Military Scenario Definition Language (MSDL).*

MIL-STD-2525C. 2008. *Common Warfighting Symbology*.

NATO Allied Command Operations (ACO), 2013. *Comprehensive Operations Planning Directive (COPD), Interim Version 2.0*, dated 04.10.2013.

Vieira, Jr., H. 2012. "NOB_Mixed_512DP_template_v1.xls design spreadsheet." http://harvest.nps.edu/ [Accessed March 31, 2015]

Zanella, J. 2012. *Combat Power Analysis is Combat Power Density*. School of Advanced Military Studies, United States Army Command and General Staff College, Fort Leavenworth, Kansas

AUTHOR BIOGRAPHIES

DANIEL HUBER is a Scientist at the Fraunhofer Institute for Intelligent Analysis and Information Systems at Sankt Augustin, Germany. He received his doctorate in Business Computing from the University of Paderborn in 2009 and was awarded a diploma in Industrial Engineering in 2004 from the same institution. He received a scholarship from the Deutsche Forschungsgemeinschaft and worked in 2012 on model simplification by using hybrid models at Arizona State University. His main research interests are modelling and simulation methodology, model simplification and planning and simulation of production systems and military operations. His email address is daniel.huber@iais.fraunhofer.de.

DANIEL KALLFASS is a project manager for M&S related projects and studies at Airbus Defence and Space GmbH in Immenstaad, Germany. He was awarded a diploma in Computer Science from the University of Karlsruhe, Germany in 2005. His main focus is on the development and application of 3D simulation systems such as PAXSEM in conjunction with the Data Farming methodology and distributed simulation federations for national and international simulation based analyses. He is involved in the German VIntEL simulation testbed demonstrator, participated at the NATO MSG-088 and is currently a technical member of the NATO MSG-124. His email address is daniel.kallfass@airbus.com.