# Simulation-based Defense Planning

**Dr Farshad Moradi, Dr Johan Schubert**
Department of Decision Support Systems
Division of Information and Aeronautical Systems
Swedish Defence Research Agency
SE-164 90 Stockholm
SWEDEN

farshad.moradi@foi.se, johan.schubert@foi.se
http://www.foi.se/fusion

## *ABSTRACT*

*This study is focused on a multi-criteria simulation-based approach for defense planning, where the concern is the best use of available resources in military ground warfare operations. As the number of possible combination of parameter values is extremely large we use a data farming approach. The scenario is simulated using the FLAMES simulation framework. We analyze large amounts of output data from the simulations in order to find the best parameter values. Since the quality of a plan is determined by several measures of effectiveness, it may be difficult to directly determine which plan is best. Often the desired result for all criteria cannot be achieved. We use preference analysis where decision makers' may indicate which criteria are most important. Based on these preferences we can determine which plan best meet all criteria. Finally, we develop methods that demonstrate why the best plan is good.*

## 1.0 INTRODUCTION

In this paper we describe the modelling, simulation and data analysis of a multi-criteria simulation-based approach for defense planning. With this method a decision maker can analyze alternative scenarios, where the concern is the best use of available resources in military operations. We evaluate tens of thousands of alternative ground warfare plans with different settings using multiple measures of effectiveness that measure how well each individual plan performs. The basic scenario represents a limited ground combat situation and describes military units and their capabilities. Using war-gaming we investigate how a combat sequence in a typical scenario proceeds and can be described in terms of simulation parameters. As the number of possible combination of parameter values is extremely large we use a data farming approach. This allows us to manipulate the simulation models through experiment design where the different parameters in the simulation are varied in a systematic way, while limiting the number of simulations. The scenario is simulated using the FLAMES simulation framework. We develop a scenario model that represent various military units and contain basic logic for various components of each unit, such as platforms, sensors, cognition models governing the behavior of different units, etc. We analyze large amounts of output data from the simulations with different values of the input data in order to find the best parameter values for the blue side, e.g., number of red and blue platoons, red and blue sensors ranges, red and blue force tactical behavior, and avenues of approach. Since the quality of a plan is determined by several measures of effectiveness, it may be difficult to directly determine which plan is best. A particular plan might be good in minimizing blue losses, another plan might be better from the point of view that the red side fails its objective. Often the desired result for all criteria cannot be achieved. We use preference analysis where decision makers' may indicate which criteria are most important. Based on these preferences we can determine which plan best meet all criteria. Finally, we develop methods that demonstrate why the best plan is good. With this methodology it is possible to find which combination of parameter ranges that leads to overall blue success.

In section 2 we describe the data farming process, and present how it is used in this paper. In section 3 we continue by describing the defense planning scenario that we investigate in this study. The modelling and simulation approaches used are described in section 4. After this, the data analysis methods and the decision support it provides is presented in section 5. Finally, conclusions are drawn (section 6).

## 2.0 DATA FARMING

Data farming [1] is a process that can be said to be a combination of already existing processes and technologies that make up a tool suite to maximize the information available. The focus is on trying to produce a sufficiently complete landscape of potential outcomes, and identify areas of special significance, rather than identifying an individual response [2]. In addition to identifying significant effects and relationships between the factors, great importance is also placed on detecting possible anomalies and including them in the decisions.

Data farming aims to provide insights into the problem formulations and is an iterative process consisting of a "loop of loops", see Figure 1.

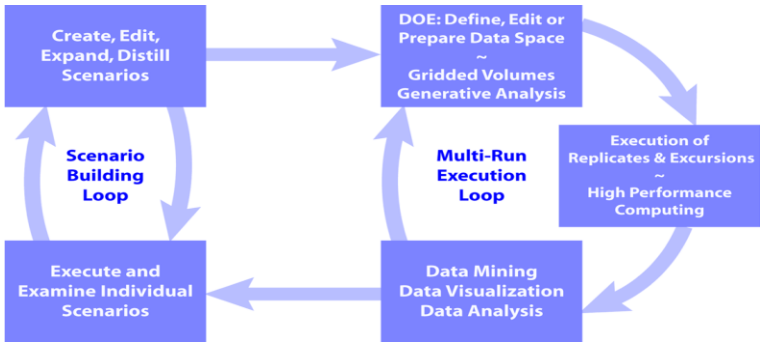


**Figure 1: Scenario development and experimentation loop.**

There is perhaps no optimal decision in a system where there are opponents acting on its own mind, but since the decision maker is allowed to understand the landscape of possibilities the notion is that more informed decisions can be made.

Based on the characteristics of the problem to be solved, there is a need for modeling nonlinearities, abstractions and influence between the various parts in a functional way. It is the combination of simple, efficient, and abstract models, as well as high performance computing, together with effective experiment design that enables quick exploration of a solution space. Simple models make it easier to manage a large number of simulation runs, which enables exploration of a large parameter and value space, and allows for the investigation of the solution space. The result is a landscape of output which can then be used to analyze trends, detect anomalies and to generate insights about multiple parameter dimensions. In addition to identifying the General characteristics the analysis also strives to give understanding of the spread and central tendencies as well as elucidate internal parameter relationships and thresholds.

The core of data farming is based on a rich and diverse array of different simulation runs that are carried out on supercomputers to check different assumptions, to gain new insights into relevant relationships, as well as to get more robust statements on opportunities and risks in specific mission situations. This is achieved by systematically varying the different parameter values for the input parameters that are assumed to be crucial as a measure of effectiveness.

As the problem we intend to simulate contains 32 binary parameters and three continuous the number of

possible simulations become 4.3 x $10^{12}$ if we assume 10 options for the continuous parameters. For a PC with 12 processors take this about 4900 years to complete. Instead we let data farming make a selection of 1 000 000 simulations which can be simulated in 10 hours on the same PC.

With data farming, we can manipulate the simulation models to our advantage through experiment design because the simulation of every value and combination of values is not feasible – despite high performance calculations. The choice of experiment design limits the information that can be extracted from the model which stresses the importance of filling up the parameter space as efficiently as possible.

Latin hyper cubes are an experimental design that focuses on many factors with complex models. Since these can also be combined with more classic experiment design to tailor the design based on the problem in question it is often used in data farming. The strength of using Latin hypercube design is that the design works especially well when a priori knowledge regarding the factors response is low. Some other strength is its effectiveness, space filling properties, and design and analysis flexibility. Orthogonal and nearly orthogonal Latin hyper cubes (NOLH) have advantages when it comes to match the model to the data. By using NOLH the model's factors are guaranteed to be un-confounded. This experimental design achieves an exploration without major cavities and it becomes possible to identify the dominant factors and cater to non-linear behaviors.

NOLH designs are important in a number of ways, they are very effective in the number of computer runs ($n$) needed to examine the number of parameters ($k$) that are of interest. Since these designs usually have good space filling and orthogonality, they are an attractive alternative to explore an unknown response function. If the simulation model contains a mixture of continuous and discrete parameters which can assume a different number of values per parameter, we may use a method that extends the methodology for constructing NOLH to manage discrete elements. The model also includes the categorical parameters; we can use a methodology called nearly orthogonal, nearly balanced mixed design (NONB).

In order to obtain maximum statistical significance in the output and allow for good conclusions over 17 input parameters on the Blue side which are control parameters in the defense planning context we are choosing to construct two separate NONB-designs for the Blue and Red side. From this we construct a complete design for all input to our simulation tool by crossing these designs, taking all instances of input parameters on the Blue side, and simulating them against all instances of input parameters on the Red side.

For the Blue side we construct a NONB with 50 alternative instances of input parameters for all 17 Blue parameters ($n$ = 50 and $k$ = 17). In order to obtain a larger NONB we are building four copies of this NONB through permutations. We stack these five NONBs and get a larger NONB for 250 alternative simulations of the Blue page parameters. For the Red side designs a NONB with 50 alternative instances of input parameters for the 18 Red parameters ($n$ = 50 and $k$ = 18). Here, we choose to build three copies of the Red NONB through permutations. In the same way we stack these four NONBs for the Red side and get a full NONB for 200 alternative simulations of the 18 Red parameters. A simple rule of thumb is that a NONB for control parameters (the 17 blue as we assume to control) should be greater or equal to the NONB for the 18 parameters that the opponent controls.

We combine the Blue and Red NONBs to get a full design for all input parameters, i.e., we take all 250 alternative instances of input parameters on the Blue side against all 200 alternative instances of input parameters on the Red side. We get a design for 50 000 alternative instances of input parameters for all 35 scenario parameters (we have $n$ = 50 000 = 250 x 200 and $k$ = 35 = 17 + 18). Finally, we run the 50 000 alternative instances of input parameters 20 times each, since the simulator use stochastic processes and we must run each instance of input parameters several times in order to be able to rely on the results. This gives us statistical significance in the subsequent data analysis. In total we run 1 000 000 (= 50 000 x 20) simulations.

## 3.0 SCENARIO

One of the initial tasks of the project is to investigate how a combat sequence in a scenario proceeds in order to set the foundation for developing the simulation model. This is done using war gaming [3], which helps us to come up with the key parameters and decision situations. We begin the process by defining and agreeing on different concepts such as scenario and events. Next, we break down a game situation in various events in order to come up with different ways to identify parameters, decision rules and actions. Based on this we develop a scenario using war gaming on a map. We analyze the course of the game; followed by suggestions on how the scenario and the identified decision situations can be described in terms of simulation parameters. Since we plan to deploy the data farming method in this work, we need to be able to create different versions of the scenario to act as input to our simulation runs. Hence, when modelling the scenario we need a set of general variables that can be used for all versions of it and a set of scenario-specific variables that depend on the current scenario terrain, types of units, tasks, and situations requiring tactical decisions.

To model the scenario in this project, it is important to broaden the space of possible decisions that the simulated unit leaders can make, and the actions that the simulated units can take. Again, the reason is to not miss any possible and interesting situations when running simulations using a data farming approach. However, we initially choose mainly the decision and policy options identified during a portion of the war game. The idea is that in the future as the project proceeds forward, the scenario model is enhanced until the entire course of the game with all the actors can be modelled. Continued games will also provide additional data for the parameters that are important to simulate and the interactions between different actors that may be relevant to deal with in the simulation.

Typically speaking a scenario in context of the defense planning games is a description of the roles of different actors and their activities extended over a long period of time and over large areas. We initially focus on a smaller part of a scenario, as explained earlier, where only a few actors are active. This is called a *vignette* which consists of a number of events together with actors who perform some specific activities, such as moving forward, reconnaissance, opening fire, etc.

In dialogue with the client, the project has agreed that an initial vignette should include a limited ground combat situation. Hence, the vignette used in our work unfolds in the context of defense against armed attack, where an attack has been going on for a number of days before this happens. In this situation the red forces has air-dropped a parachute battalion (1st Fskbat) at an airport and are in battle with a mechanized battalion (1st Mekbat) reinforced with a tank company (Strvkomp).

A new airdropping of a 2nd Fskbat (red forces) occurs in the areas around Gimo (see figure 2). The task of this unit is to move forward towards the airport in order to support the 1st Fskbat in ensuring that the airport stays open for landing of transport aircrafts, carrying new military units. At the same time a 2nd Mekbat (blue forces) has regrouped and is positioned in northern part of Uppsala. The task of this battalion is to prevent red forces from reaching the airport.


**Figure 2: Scenario map.**

This part of the scenario (2nd Fskbat vs. 2nd Mekbat) is the focus of our simulation. Organization of the forces (Order of Battle - ORBAT) is as follows, a mechanized battalion on the blue side and a parachute battalion on the red side. In this example, these units are organized as described in figure 3, respectively figure 4. The units are greatly simplified for several reasons. As previously explained the main reason is that the project should initially have a manageable number of actors to consider. As work progresses, the vignette will evolve with more types of units which will also include other types of systems.

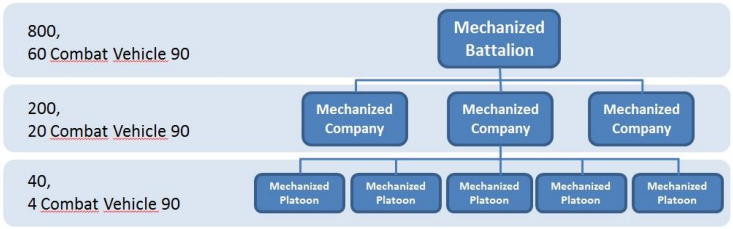


**Figure 3: Blue order of battle.**

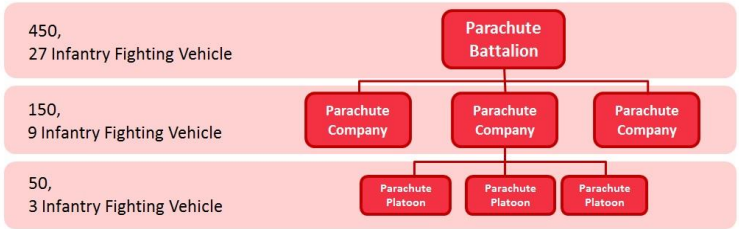


**Figure 4: Red order of battle.**

The next step of this work is to describe the scenario in terms that can easily be adapted to our simulation

tool. Furthermore we describe what values (states) the different parameters can be assigned.

# 4.0 MODELING AND SIMULATION

## 4.1 Modeling

One of the main requirements of the project is that the defined scenario should be suitably and rapidly modeled and simulated such that it answers the questions that are relevant to the defense planning. Based on this requirement, already in the early stages of the project it is decided to use the FLAMES (Flexible Analysis and Mission Effectiveness System) simulation framework in our work. The FLAMES simulation framework [4] is primarily used for simulation of complex military scenarios at the combat/tactical level. The tool is modular and provides general simulation functionality such as time management, database management, scenario generation, visualization and analysis. It also contains a number of ready-made models of various military units on different levels, which is a good starting point and can reduce development time.

The main aspect of the modelling work is to translate the defined scenario in terms that is understandable by FLAMES. The scenario as explained in section 3 is at tactical level and consists of 15 red platoons and 15 blue platoons. The blue platoons include 4 “Combat Vehicle 90” units and the red platoons 3 BMP-1 (Infantry Fighting Vehicle) units. Each platoon also consists of a leader and the rest, so called “wingmen”. The reds are located in the north at various points, while the blue ones are located in different areas along the different routes to the south. The objective of the scenario is that the Reds try to get past the defense line located at about the same level as the route 77 to the south, see figure 2.

In order to build a simulation model in FLAMES based on the scenario, we use a map of Sweden (in GeoTiff) with elevation data (DTED), and develop models describing the basic logic for different classes of objects, such as platforms, sensors, cognitive models, etc. Generally, when building models (components) in FLAMES, one can start with some basic models that are already available, which can then be extended to fit one’s own needs. In our case, we have developed some special models to manage more complex cases that take place in the scenario. These models include three cognition models (SWECon, BMPTank and S90Tank), two platform models (Stridsfordon90 and BMPTankPlatform) and a formation model (SWEGroundFormation), see Figure 5.

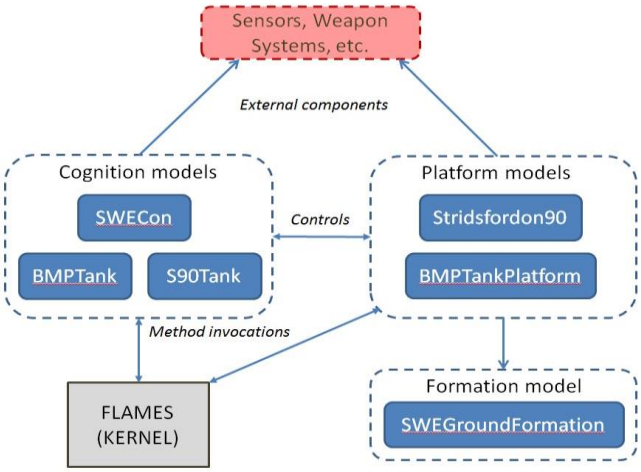


**Figure 5: Cognition, platform, and formation models.**

Cognition models are used for simulating human behavior and cognitive processes, while platform models are deployed for simulation of man-made devices and equipment, such as sensors, missiles, vehicles, ships, munition, etc. The formation model, as the name implies, contains logic about placement of the entities.

### 4.1.1 SWECon

One of the main requirements of the project is that the defined scenario should be suitably and rapidly modeled and simulated such that it answers the questions that are relevant to the defense planning. Based on this requirement, already in the early stages of the project it is decided to use the FLAMES (Flexible Analysis and Mission Effectiveness System) simulation framework in our work. The FLAMES simulation framework is primarily used for simulation of complex military scenarios at the combat/tactical level. The tool is modular and provides general simulation functionality such as time management, database management, scenario generation, visualization and analysis. It also contains a number of ready-made models of various military units on different levels, which is a good starting point and can reduce development time.

SWECon is a cognition class, which acts as a "logger" that records various events during the course of the simulation. What happens in practice is that SWECon polls FLAMES-kernel for all available data on all devices. Data of interest include:

- Simulation time,
- Total number of units,
- Total number of blue units,
- Number of blue entities that are alive,
- The total ammunition for blue,
- Ammunition left blue,
- Total number of red units,
- Number of red units that are alive,
- The total ammunition for red,
- Ammunition left red,
- Number of red crossed line of defense,
- Route selected by the red force.

After the simulation is complete, SWECon prints out the information (i.e. collected data), which can then be used by other programs for analysis. In SBFP SWECon can be seen as the brain that provides information on what we are interested in. It is also capable of reporting damages. For this reason a "Damage" method is implemented in "BMPTankPlatform" and "Stridsfordon90" platforms.

### 4.1.2 S90Tank

S90Tank is another cognition class based on an existing tank class in FLAMES. In addition to the existing functionality we include properties to enable activation of the platform, customization of various attacking behavior, inching behavior and reporting of damage to the SWECon. S90Tank-cognition model, along with the driver's cognitive model, act as the brain of a Combat Vehicle 90. Any changes regarding the behavior of the Combat Vehicle 90, such as how to respond to enemies is made in this class.

### 4.1.3 BMPTank

Just as S90Tank this cognition class, along with the driver's cognition class, act as the brain of the enemy's combat vehicles of type BMP-1. However, they lack the special behavior for inching and ambushes.

### 4.1.4 Stridsfordon90 (platform)

This class serves as the platform for the Combat Vehicle 90. For a device to appear in FLAMES it must have a platform and for the platform to be shot at, it has to have a signature. Stridsfordon90 platform provides information on how the vehicle can move around, including which advance behavior the vehicle has. This platform also calculates the damage done to it.

Stridsfordon90 platform is coupled with S90Tank-cognition model in one "Pattern". This then enables the Combat Vehicle 90 to be selected as a "Unit" in FLAMES.

### 4.1.5 BMPTankPlatform

This class is the platform for BMP-1, which is then connected to BMPTank-cognitive model in a "Pattern". The only thing that is extended in this class apart from the basic functionality is the ability to calculate the damage to the platform.

### 4.1.6 SWEGroundFormation

SWEGroundFormation is a formation model that includes logic for how entities are placed e.g. in a platoon. In this formation model, we have implemented a logic in which the entities are placed randomly behind a leader based on the leader's position.

## 4.2 Simulation

In SBFP we analyze the simulation results not based on a single run, rather hundred thousands of runs. We do this by creating many outputs based on a defined amount of input data. For this purpose we develop many different variations of the same scenario by varying the input parameters (scenario variables), thus we will have a better statistical certainty. The whole process is called Data Farming in which we actually "grow" a huge number of possible outputs based on a large combination of input data. Table 1 depicts the input data which we have defined for the simulation and the range of values they can assume.

The challenge is that input data must be selected in such a way so that there is not a combinatorial explosion. To support data farming we have created programs which depend on FLAMES running the scenarios in batch mode. These programs first produce an input file containing permutations of our basic scenario, which is done by randomizing between the minimum and maximum values of each scenario variable. These permutated scenarios are then fed to our simulation. Later the results are gathered in an output file generated by the SWECon model, and verified against the input file. Here we can check whether an error has occurred, and what the error is about. If there are no errors the results are delivered to the analysis program. In order to reduce the execution time the data farming scenarios are run in parallel.

In our experiment we have 50 000 permutated scenarios, where each scenario is run 20 times with different seeds, making a total of 1 000 000 runs. When deploying a standard desktop with 12 processors, the experiment takes about 10 hours to complete.

**Table 1: Scenario values.**

| Scenario variables | Values |
|---|---|
| Number of blue units that are active | 1..15 |
| Sensor range for the blue forces | 1500..2500 m |
| Firing behavior for the blue forces | 1 = flames default, 2 = multi-target-mode |
| Sensor range for the red forces | 1500..2500 m |
| Speed of the red tanks | 8..45 mph |
| Number of the red units that are active | 1..15 |
| Routes that the reds can take | **A**, **B**, **C** (3 different routes with the coordinates given below) |
| (waypoint(x)_lat, waypoint(x)_long) | **A** = [(59.783598, 18.138005), (59.775169, 18.134743), (59.766412, 18.14192), (59.763128, 18.139527), (59.752291, 18.148227), (59.739812, 18.157796)]<br><br>**B** = [(59.785569, 18.228042), (59.769696, 18.223475), (59.761924, 18.208904), (59.753823,18.224562), (59.745176, 18.205859), (59.741235, 18.204336)]<br><br>**C** = [(59.779876, 18.248702), (59.772214, 18.242178), (59.766631, 18.229782), (59.754371, 18.252182), (59.746927, 18.25153), (59.73598, 18.252617)] |

## 5.0 DATA ANALYSIS AND DECISION SUPPORT

When data is collected, experiment design, and simulations are done, often large amount of data is generated – a result that can be both cumbersome and unwieldy. We are interested to find out what the results really means and what conclusions that can be drawn. This may also generate a variety of issues that need to be dealt with. An example of areas in which we may want to shed light on is:

- distribution of system response,
- central tendencies of system responses,
- relationships between system response,
- how various factors (input variables) affect each other and the system response,
- interesting areas or thresholds for factors,
- General characteristics of the landscape of possibilities.

How can information on the above points be helpful? By modeling and simulation we know what the *building blocks* looks like and what results are generated – however, we often have very little knowledge of the inner system, the dynamics, which causes the outcomes observed. Through the application of useful methods for analysis and visualization of output data our understanding may deepen and answers may be

found.

In data farming we often deploy a mixture of techniques from the fields of visualization, data mining and statistical analysis. Statistical analysis techniques are used to explore the previously mentioned points – for example, through characterization of distributions, to put up confidence intervals and implement hypothesis testing. Data mining is used to search for patterns and relationships, and visualization techniques can be powerful tools to investigate, explore, and present data. To examine data through visualization may serve as a validation method, but also for the quality assurance of data. Visualization for exploration contributes instead to find new information and new insights. Finally, visualization concerns how to convey message and understanding of how data and results should be presented so that the messages become easily comprehensible and easily accessible. Technologies and their purposes can therefore look quite different – common to all of them, however, is the starting point that a good simulation must be efficient, precise, aesthetic and adaptable.

Rather than relying on a few technologies, data farming take advantage of a combination of methods in which each technology's special strength is used in an as efficient way as possible. These areas are not completely separated, instead they overlap – both in terms of techniques and the information they seek to extract. Often, it may be a good idea to begin the analysis of a problem involving a large number of influencing factors with visualization in order to reach a broader holistic understanding. This can then be followed by the application of more statistical analysis methods and then include additional visualization – simply an interaction in which the purpose controls the selection of methods.

## 5.1 Pareto and preference analysis

Pareto analysis is a form of multi-criteria analysis. The simulation results are analyzed for both input and output, and a number of criteria that must be met are set up. Such as Blue losses should be as small as possible and as few Red units as possible reach their desired goal. The Pareto analysis states that if there *is a set of input that on all output criteria provides better outcomes than what a different set of input gives, so is the first is Pareto optimal over the other*.

In our simulation, we simulated 50 000 alternative input instances, each 20 times with different random seeds (that is, a total of 1 000 000 simulations). We have analyzed how each output parameter varies over the 20 simulations with different random seeds (e.g., how many red units reached at most or at least its objective). This gives us an idea of how much randomness can influence the outcome with the same input data.

Initially, we try to find the most effective plan instance as evaluated by all measures of effectiveness (MOEs). When there are many such measures we are faced with a multiple-criteria decision making problem when assessing which plan instances are preferred. As a first step, we develop a new method for finding the Pareto optimal frontier of the entire set of all plan instances where utility intervals over each measure of effectiveness is received from the 20 simulations, due to the stochastic nature of the simulations. The plans on the Pareto optimal frontier are the plans that are better than all other plans regardless of how the measures of effectiveness might be weighted in a subsequent assessment process.

As we assume that it is impossible for decision makers to assign precise weights to all measures of effectiveness, we let a group of decision makers express any number of preferences on the importance between any two disjoint subsets of MOEs [5][6]. We further develop an extension to Utkin's [7] preference ranking method which is focused on finding the order of importance of the measures of effectiveness from the preference assignments made by the decision makers.

Using the preference order of importance for all MOEs we develop a Monte Carlo approach for assigning weights to these MOEs. In this method we randomly assign weights that abide by the preferred order of

importance of the MOEs. That is, the most preferred measure will be weighted higher than the second most preferred measure, etc. In the spirit of the Monte Carlo approach we perform 1000 alternative weight assignments for all measures, yielding 1000 alternative rankings of all alternative plan instances. The plans with the highest average ranking over the 1000 alternative rankings are the most preferred plan instances. A process overview is provided in Figure 6.


**Figure 6: Process overview.**

In our example, it turns out that 23% of the simulation outcomes are screened out as non-Pareto optimal, meaning that they are worse on all criteria for at least one other simulation. In future we focus the data analysis on the remaining 78% of the simulations that have a crucial influence on the outcome of the subsequent data analysis.

## 5.2 Decision support

A first overview of input and result variables is provided by the histogram that shows how often the variables assume different values at the given interval. Note that we are measuring the values that different parameters takes over 1 000 000 simulations. Thus, we do not measure what values a parameter takes given some fixed input. A fact that soon becomes obvious based on the histograms is that the amount of ammunition is not a limiting factor for either side. Figure 7 shows the histogram of the percentage of remaining ammunition for the Blue and Red side. Neither side ever spends more than one-third of available ammunition and typically less than 5% is spent. Thus, ammunition variables are seen as insignificant (given this model) and therefore are ignored in the subsequent analysis.

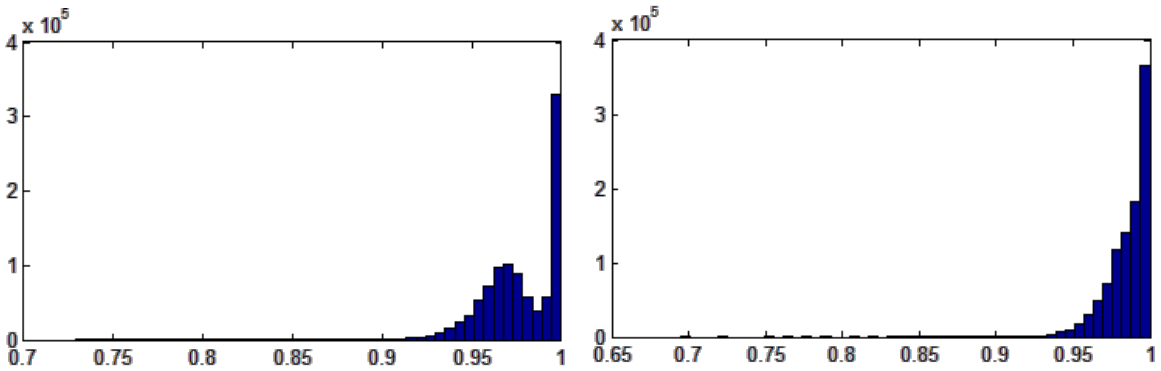


**Figure 7. Histograms of percentage remaining ammunition for the Blue (left) and Red (right).**

Looking at other result variables (Figure 8), we see that the most common individual cases are *no loss*, and that *no red reaches the finish line*, but variables are also distributed over other values.

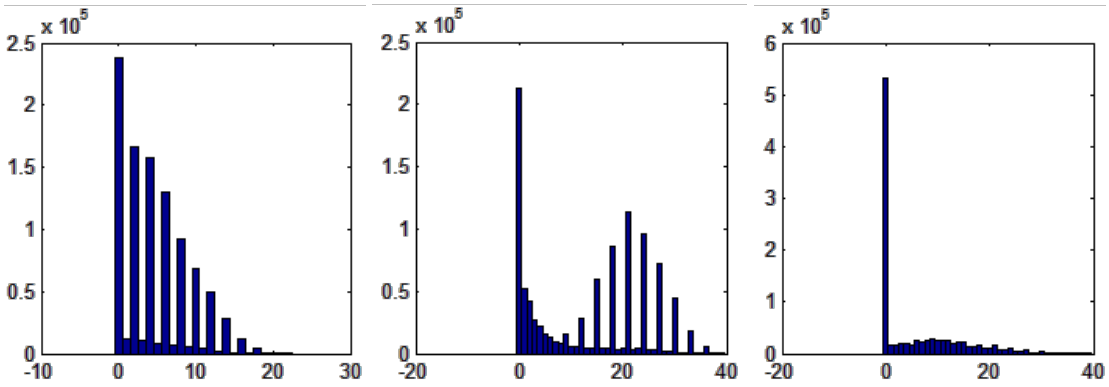


**Figure 8. Histogram of the number of Blue and Red losses, as well as the number of Red successful, respectively.**

To study how the balance of power between Blue and Red, in terms of number of platoons and sensor ranges, affect the outcome we draw up heat maps of losses and number of successful red units as a function of number of platoons on either side as well as sensor ranges on each side, see Figure 9, where dark and light represents the high and low values in the resulting dimension.

We see a certain tendency that more platoons on one side leads to more losses on the same side. It also appears to be a remarkably weak trend to more losses when the opposing side is large. Sensor ranges exhibit a clearer picture. Each side is favoured strongly by having longer ranges than the opposing side, both in terms of losses and finishing statistics.

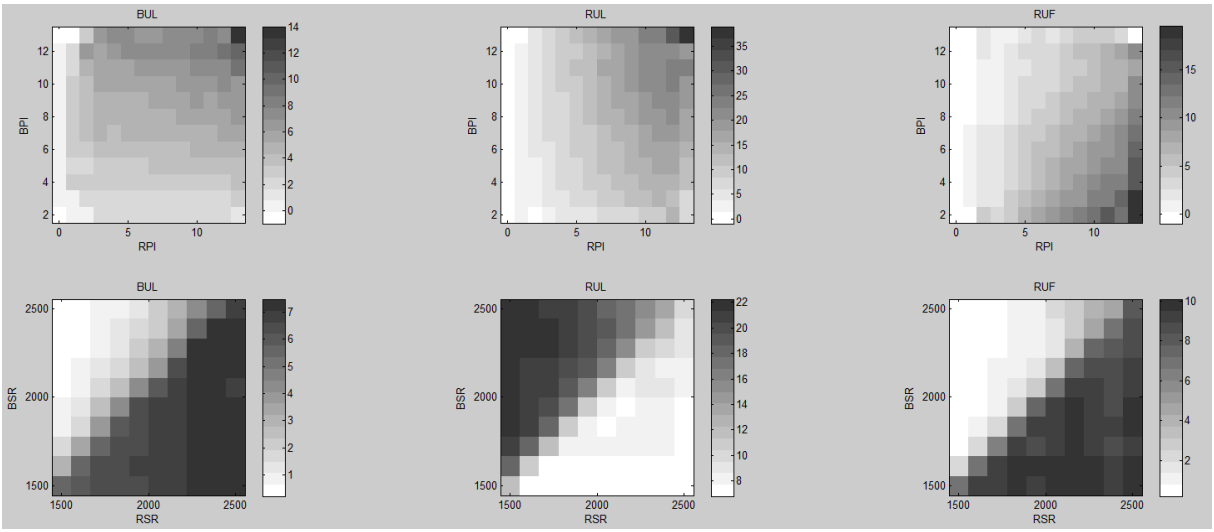


**Figure 9. Grayscale charts that illustrate the outcomes as a function of strength and ability between Blue and Red. *Upper row*: number of Blue (BPI) or Red (RPI) units. *Bottom row*: sensor rages for Blue (BSR) or Red (RSR) side. *Column 1*: Blue losses (BUL); *column 2*: Red losses (RUL); *column 3*: Red that are successful (RUF).**

In order to further study how the finishing statistics of Red looks in proportion to the number of participating platoons on each side, we draw up box plot charts, see Figure 10, where the result variable distribution is displayed for each value on the input variable, where median is designated as a red line and the span between the quartiles closest to the median is marked with a blue box. Here we see that when the number of blue platoons exceeds six the likelihood that some Red will finish successfully drops to below 50%. The same

applies when the number of Red platoons is less than 10.

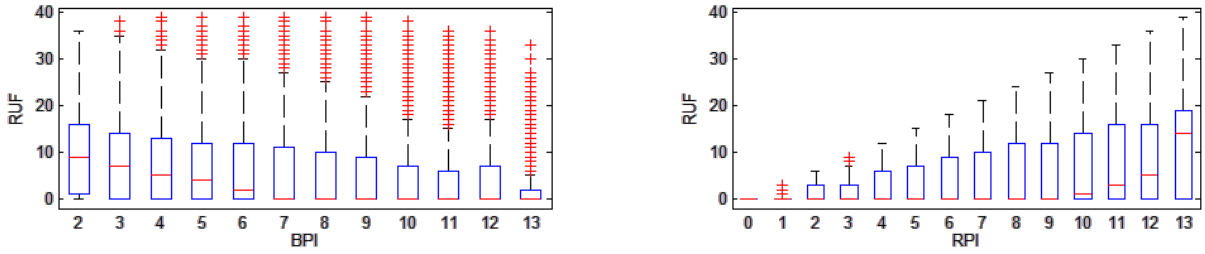


**Figure 10. Box plot charts indicating how many Red units that finish varies with the number of participating Blue (left) and Red (right) platoons. *Left picture*: the number of Reds that finish as a function of the number of Blue units involved; *Right picture*: the number of Reds that finish as a function of the number of Red units involved.**

We can also study the impact of the categorical tactics variables BFB (Blue fire behavior) and RW (Red avenue of approach) has on the output dimensions using box plot charts, see Figure 11.

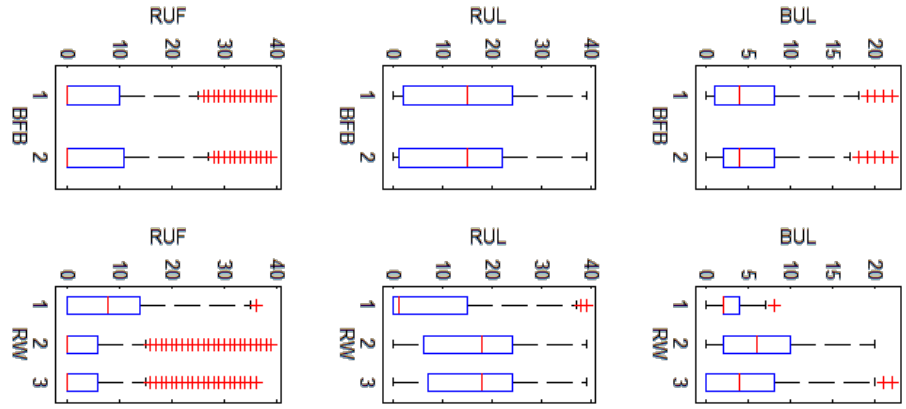


**Figure 11. Box plot charts over output (BUL, RUL and RUF) as a function of tactics variables BFB and RW. *Upper row*: Blue tactics (BFB) "1" = shoot immediately, "2" = lie in ambush. *Bottom row*: Red avenue of approach (RW) "1" = circumvention, "2" = attack route 2, "3" = attack route 3. *Column 1*: Red that finish successfully (RUF); *column 2*: Red losses (RUL); *column 3*: Blue losses (BUL).**

Here it becomes clear that the BFB have only marginal impact on the outcome, while the RW variable has a larger significance: choosing avenue "1" for approach appears to be a winning strategy for Red, as it seems to lead to many more Red finishing successfully than the other two routes by avoiding confrontation (i.e., by bypassing the Blue forces).

Regression trees are equivalent to decision trees for continuous variables. It is a method suitable to predict the outcome of any result dimension based on given values on input. This allows the tree to give a picture of which input variables that are most significant for the outcome. A regression tree for result variable RUF (Red case) shown in Figure 12.

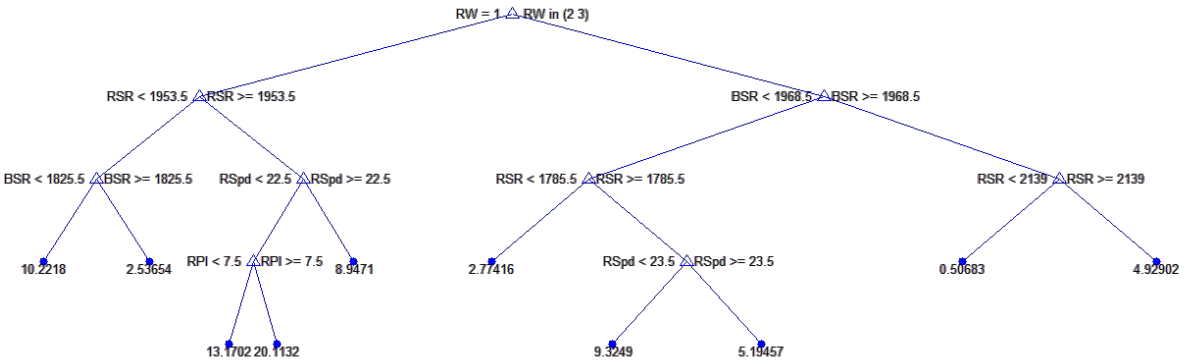


**Figure 12. Regression trees, with which the RUF can be predicted based on given values on input variables. For a variable appearing high up in the tree indicates that it is of great importance for the output value.**

Here it becomes clear that the Red approach road (RW), as previously observed, is crucial to the outcome of the simulation. In addition, sensor ranges (BSR/RSR) and Red speed (RSpd) are important predictors. With the help of the regression tree the input variables relative predictive importance can be quantified, as shown in Figure 13.

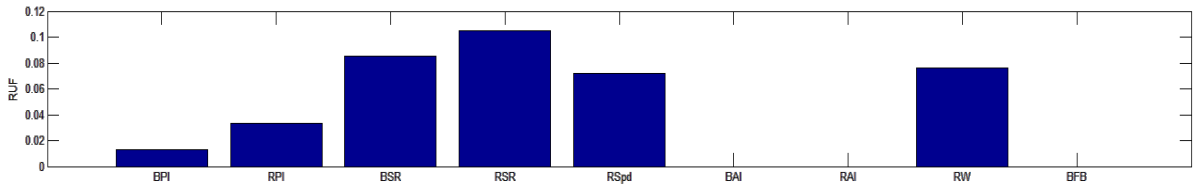


**Figure 13. The input variables relative importance for prediction of the number of Reds that finishes successfully.**

Sensor ranges (BSR, RSR), Red avenue of approach (RW) and Red speed (RSpd) dominates; second are the number of Red and Blue platoons (RPI, BPI). The image matches well with the message from the other figures.

## 6.0 CONCLUSIONS

We have developed a simulation-based decision support system for analysis of alternative situations in defense planning. With this system we can simulate situation examples using a data farming approach that systematically draws samples by varying all input parameters. We have developed a multi-process multiple-criteria decision making methodology for assessing the large amounts of output data from the simulation. The alternative simulations are evaluated by several different measures of effectiveness. The decision support methodology uses Pareto analysis followed by preference analysis of measures of effectiveness and Monte Carlo weighting of the measures within the given preference order. With this methodology we first find the most important input parameters that lead to Blue forces success. Secondly, we investigate parameter values and parameter combinations to explain the causes that lead to Blue success.

## 7.0 REFERENCES

[1] Horne, G.E., Meyer, T.E. (2005), Data farming: Discovering surprise, in M.E. Kuhl, N.M. Steiger, F.B. Armstrong, J.A. Joines (eds.), in *Proceedings of the 2005 Winter Simulation Conference*, Orlando, FL, 4–7 December 2005, pp. 1082–1087.

[2] Horne, G., Åkesson, B., Anderson, S., Bottiger, M., Britton, M., Bruun, R., Choo, C. S., Erdoğan, O., Ergün, İ. Y., Geiger, A., Gremmelspacher, D., Hartmann, J., Hölscher, M., Kallfass, D., Lappi, E., Maly, A., Mässeli, K., Mayer, S., McDonald, M., Meyer, T., Narayanan, F., Ng, E. C., Ng, K., Pakkanen, M., Sainio, J., Sanchez, P., Sanchez, S., Schubert, J., Schwierz, K.-P., Seichter, S., Upton, S., Wagner, G., Wan, S. C., Whitney, L., Yiğit, A., Yıldırım, U. Z. and Zimmermann, A. (2014), Data Farming in Support of NATO – Final Report of Task Group MSG-088, STO Technical Report STO-TR-MSG-088, NATO Research and Technology Organisation, Neuilly-sur-Seine, France, 2014. [Online]. Available: http://ftp.rta.nato.int/public//PubFullText/RTO/TR/STO-TR-MSG-088///$$TR-MSG-088-ALL.pdf

[3] Dunnigan, J., The Complete Wargames Handbook: How to Play, Design and Find Them, Revised edition, William Morrow, 1992. ISBN 0-688-10368-5.

[4] FLAMES Simulation Framework, http://www.ternion.com.

[5] Schubert, J., Hörling, P. (2014), Preference-based Monte Carlo weight assignment for multiple-criteria decision making in defense planning, in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain. IEEE, Piscataway, NJ, paper 189.

[6] Schubert, J. (2014), Partial ranking by incomplete pairwise comparisons using preference subsets, in *Belief Functions: Theory and Applications 3*, F. Cuzzolin (Eds.), *Proceedings of the Third International Conference on Belief Functions*, Oxford University, UK, 26–28 September 2014. Springer (LNAI 8764), Berlin, 2014, pp. 191–199.

[7] Utkin, L.V. (2009), A new ranking procedure by incomplete pairwise comparisons using preference subsets, *Intelligent Data Analysis*, **13**(2):229–241.