Simulation-based Decision Support for Effects-based Planning☆

Johan Schubert, Farshad Moradi, Hirad Asadi, Pontus Hörling, Eric Sjöberg
Swedish Defence Research Agency
SE-164 90 Stockholm
Sweden
schubert@foi.se

***Abstract*—In this paper we describe decision support and simulation techniques to facilitate effects-based planning. By using a decision support tool, a decision maker is able to test a number of feasible plans against possible courses of events and decide which of those plans is capable of achieving the desired military end state. The purpose is to evaluate plans and understand their consequences through simulating the events and producing outcomes which result from making alternative decisions. Plans are described in the effects-based approach to operations concept as a set of effects and activities that together will lead to a desired military end state. For each activity we may have several different alternatives. Together they make up all alternative plans, as an activity tree that may be simulated. Simulated plans that are similar in both their structure and consequence are clustered together by a Potts spin neural clustering method. These plans make up a robust set of similar plans that function as ready alternatives should dynamic replanning be necessary as the situation evolves.**

***Keywords*—simulation, decision support, operational planning, effects-based planning, EBP, effects-based approach to operations, EBAO.**

I. INTRODUCTION

In this paper we develop decision support and simulation techniques to facilitate effects-based planning (EBP). By using a decision support tool, a decision maker is able to test a number of feasible plans against possible courses of events and decide which of those plans is capable of achieving the desired military end state. The purpose is to evaluate plans and understand their consequences through simulating the events and producing outcomes which result from making alternative decisions regarding activities. We extend previous work [1], with more detailed agent modeling and decision support methodology.

The methods developed can be used in an incremental manner by testing the plans as they are developed step-by-step and new activities are added. The plans are described in the effects-based approach to operations concept, as a set of effects and activities that together will lead to a desired military end state.

Actors and activities are modeled using a scenario used by the Swedish Armed Forces in their Combined Joint Staff Exercises. The activities of the plan are simulated together with all actors and their reactions on our planned activities, and their possible follow-on interactions. As the activities may each have several different alternatives in which manner they can be carried out, together they span-up an activity tree. The tree is searched by an A*-algorithm [2, 3] where each level in the tree is an activity and each node in the tree is an alternative for an activity. As the activity tree is searched, each node (*i.e.*, sequence of alternatives leading from the root of the three to this node) is evaluated by the simulator and results are stored and communicated to the decision support side of the system. By using an A*-search to guide the tasks of the simulator we let the simulator work in a manner to achieve maximum information value gain. In addition, a simulation control interface lets the decision maker put constraints on the search, in order to simulate activities within his area of interest.

Simulated plans that are similar in both their structure (*i.e.*, mostly the same alternative chosen for each activity) and in their consequences are clustered together by a Potts spin [4] neural clustering method [5, 6]. These plans make up a robust set of similar plans that function as ready alternatives should dynamic replanning be necessary as the situation evolves.

In Sec. II we introduce the concept of EBP. In Sec. III we describe the scenario we use in our modeling experiments. In Sec. IV we develop methods for decision makers to focus the attention of the simulator onto an area of interest. We continue by developing generic models for actors and activities (Sec. V). In Sec. VI we describe the simulation approach we developed for simulating many alternative operational plans. We develop decision support methodologies for finding groups of similar robust plans based on the simulation results (Sec. VII). Finally, conclusions are drawn (Sec.VIII).

II. EFFECTS-BASED PLANNING

An effects-based approach to operations (EBAO) [7] is a military approach for the management and implementation of efforts at the operational level. In an article by Hunerwadel [8] the author explains the concepts. Hunerwadel points out that EBAO primarily has to do with effects-based thinking. According to the United States Joint Forces Command (USJFCOM) EBAO are "operations that are planned, executed, assessed, and adapted based on a holistic understanding of the operational environment in order to influence or change system

☆This work was supported by the FOI research project "Real-Time Simulation Supporting Effects-Based Planning", which is funded by the R&D programme of the Swedish Armed Forces.

978-1-4244-6587-3/10/$25.00 ©2010 IEEE

behavior or capabilities using the integrated application of selected instruments of power to achieve directed policy aims" [9]. According to USJFCOM an effect represents a so called PMESII[1] state that results from one or more military or non-military actions.

Hunerwadel argues that military decision-makers attain objectives in order to achieve the desired end state; policy sets boundaries that limit strategies. He finishes by pointing out that EBAO is a thought process, a number of concepts and a way of thinking: "The soul of 'doing effects' is and will always remain 'thinking effects' ".

Compared with previously employed views from leading military quarters a new approach is voiced (*i.e.*, EBAO), in particular regarding the requirements on the understanding of the situation and the methods that can be used to achieve political-strategic desired effects. In the words of E. A. Smith [10]: "The cognitive domain is the real focus of any effects-based operation", which may be interpreted as if the purpose of military operations is always to influence other players' perceptions and behaviors. To reach the politically desired effects far more resources and more sophisticated types of effective resources other than arms or violent means of power must be used. We must carefully analyze the effects we want to achieve before selecting the objectives and means for their strategic action.

The process of EBAO consists of four connected parts: EBP for developing plans, effects-based execution (EBE) for carrying out those plans, effects-based assessment (EBA) to follow-up on the plan execution, and knowledge support providing the other three processes with background knowledge.

A control theory model of EBP [11] is shown in Fig. 1. As input we have the required situation R_s which is compared with the current situation C_s received from assessment. The first process is an end state analysis (ESA), followed by effects development (ED). Initially when there is no operation the military end state defines the goal of the operation. Later when a campaign assessment is carried out, the comparison between R_s and C_s may require further analysis in ESA. The output from ED is the required effects R_e which is compared with the current effects C_e, also received from assessment. The next process is action development and resource matching (ADRM) followed by synchronization and plan refinement (SPR). All processes take inputs from red-green activity (RG). The output from SPR is a plan to be executed by EBE. Campaign assessment C_s is received from a qualitative campaign assessment and current effects C_e is received by measure of effectiveness and measure of performance analysis in EBA.

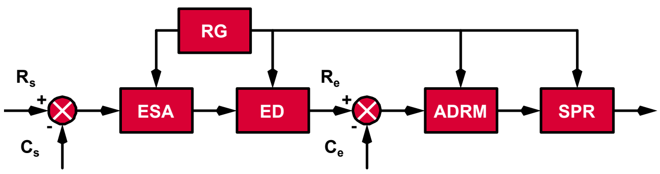

Figure 1. The processes of EBP.

[1]political, military, economic, social, infrastructure and information.

III. Scenario

We make use of the same scenario that has regularly been used by the Swedish Armed Forces in the Combined Joint Staff Exercises. The scenario comprises several fictitious countries, two of which, Xland and Bogaland, have been described in-depth. Background histories offer explanations to why and how sentiments, stances, identities, loyalties, economic dependencies and inequalities have evolved over time, occasionally resulting in shifts of power. Phenomena that are commonly found in conflict areas and post conflict areas have been embedded in scenario contexts that make the origins of the phenomena plausible.

In Xland demographic change constitutes a threat to the privileged majority group, and puts severe pressure on the government. The country has a constitution that does not give the fast growing minority group the same rights as the dwindling majority group. Irregular groups originating from the minority group have taken control of those rural parts of the country that used to supply raw materials to the biggest industries in Xland. This has resulted in a loss of revenues, environmental degradation and incentives for foreign actors to intervene in order to protect their economic interests in Xland.

In Bogaland, a newly industrialised country, a civil war broke out ten years ago when discontent within the minority ethnic-religious group had reached very high levels. The root cause was increasing social stratification caused by what members of the minority group perceived as unjust distribution of revenues from a natural resource located in an area populated by the minority group. The civil war put an end to the exploitation of the resource, in this case oil, and revenues dropped to very low levels. The country was split into two parts, roughly along ethnic lines, with each part having its own government. A post-war economy evolved over the next decade, and several irregulars and insurgents are now challenging the incumbent presidents.

The incumbent presidents have signed a peace-agreement, and an international force, BFOR, is present to support the implementation of the agreement. Irregular groups in Bogaland seek to preserve or increase their influence by undermining the efforts of BFOR, the governments or competing irregulars. Two of the neighboring countries have much at stake in the conflict, because of economic interests and shared identities with parties within Bogaland. Actors within these neighboring countries support irregulars in Bogaland.

IV. Simulation Control

The planning process we develop corresponds to the selection of a subset of activities which are chosen from a set of alternative activities. A chosen combination of alternative activities constitutes a plan. The number of plans can theoretically grow very large since each permutation of alternative activities will constitute a separate plan. Of course, in practice, many of these plans can be ruled out because of necessary conditions such as one activity that under real conditions has to be executed before another activity starts. If both activities have alternatives that start at different times, all combinations of alternatives that swap the activities in time, or make them run in parallel can be ruled out.

It is therefore necessary to give the simulator instructions on how to select combinations of alternatives it should ignore, avoid, or prefer, during simulation. Some simple ways to do this is to focus the simulators' efforts towards activities that are:

- executed in a specified geographical area,
- executed within a specified timeframe,
- influencing each other strongly in the cross-impact matrix (CIM) [12, 13].

The CIM is a matrix set up for all activities where it is specified how much they support or counteracts each other during execution due to resource conflict, *e.g.*, one activity preparing for another one, etc.

In the graphical user interface, we accordingly make these selections as a preferred area of interest in a map, a timeframe in a Gantt chart, and an activity group in a chart with activities grouped according to their inter-influencing in the CIM. Each of these three types of selections gives each activity a weight between 0.0 and 1.0. The final weight for an activity, to be used for its importance in the simulation, is the product of these three weights. Fig. 2 shows the screen of this user interface.

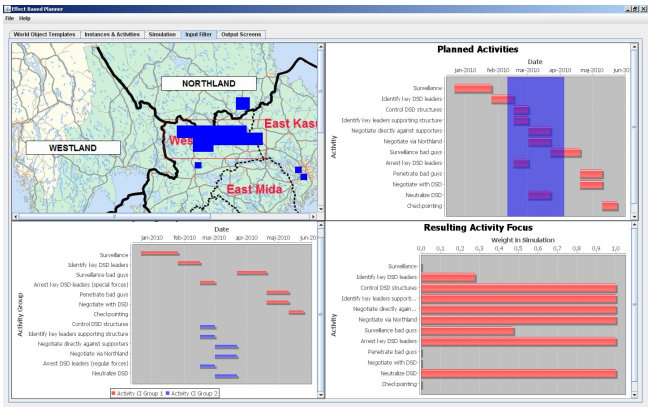

Figure 2. The simulation control tab of the GUI. Upper left: Selection of geographical focus area. Upper right: Selection of focus timeframe. Lower left: Selection of CIM connected activities. Lower right: Fused weights for the activities which give their importance in the simulation.

V. Modeling Actors and Activities

A. Activities

How we model a phenomenon depends on the purpose of the model and the questions we want to answer. Since our simulation system aims to support decision-making within EBAO the modeling has to be based on EBAO and the concepts used within it, such as plan, activity, effect, end state, etc.

A *plan* as it is defined in the context of EBAO is a sequence of *activities* that together leads to a desired *end state* which is set by a military force. These activities are *events* initiated by own forces and require different types of resources in order to be executed. Furthermore, they can affect each other and be affected by external events. External events, in our model, can either be initiated by other actors or be spontaneous/natural events. The former could either be planned, *i.e.*, an actor's action according to its agenda regardless of our activities, or responsive (dependent on our activities), such as the enemy force's response to an attack, or the local population's reaction to an operation. The spontaneous/natural events are on the other hand unpredicted incidents, such as weather conditions, natural catastrophes, an unprovoked attack or an accident.

As we can see based on the above discussion, three different types of events can be discerned in our model: the launch of an activity (our own action or an action by another actor), observations and reactions made by an actor, and external events. The main difference between these event types, as far as our model is concerned, is who is initiating the event (if there is one) and who is receiving it, *i.e.*, the target of the event. Another important aspect is whether the events are planned or responsive since that effects how the events are scheduled in our *event list*, as we will see in section VI, explaining our simulation. In case of an external event, initiation is not done by an actor.

Each event when received by an actor is interpreted as more or less hostile or friendly. This interpretation depends on the state of the involved actors and their relations, such as a degree of aversion, and is graded along a *hostility scale* from 1 (exposed to attack) to -1 (friendship strengthening initiative). Similarly, every event has a certain effect on one or several environmental objects (discussed in Sec.V.B), *e.g.*, lowers their functionality with 2 units.

Events affect actors both directly and indirectly depending on their level of involvement. The initiator(s) and the target(s) of an event are those who are directly affected by the event. Other actors may be indirectly affected based on their relationship with the initiator(s), the target(s) or the environmental objects. For instance my enemy's enemy can become my friend, or destruction of a transportation route, a power supply or a religious building may have negative effects on the local population's attitude towards our forces.

B. Actors

In our model we define an *actor* as an entity with resources, an action repertoire, an agenda and an internal state. Entities can be groups of people, who somehow have a common identity and purpose [14]. They may be more or less clearly defined and organized, and can be everything from police forces, relief agencies, well-organized militia units, and state administrative bodies to loosely coupled groups and social clusters. These loosely coupled groups and social clusters are usually held together by one common interest (which at the moment is in the focus). In exceptional cases, the actor might even be a single individual, such as a prominent opinion maker, political leaders or a financial potentate. A special actor is "we", *i.e.*, the blue force that is using this simulation.

The set of possible actions that an actor is capable of performing is called an *action repertoire*. It is determined by the actor's resources and knowledge, and that which is ideologically desirable but not yet possible for the actor to achieve. Depending on the ideology and strategy many of the possible actions are extremely unlikely because they would be counterproductive and not good for the actor's image. However, as the state of the actor changes based on the events and other actors' activities, certain actions in its action repertoire become more probable and others less probable.

The *agenda* is the plan that an actor is supposed to follow in order to achieve its goals. The state can be defined as a combination of resources, mood, solidarity, short-term agenda, etc. The states of the actors changes as a response to activities and events, together with the probability of performing different actions. Hence, each action has a probability associated with it, which is changed according to some functions.

It should be noted that the probabilities of performing actions and the agenda are not directly affected by the other actors' actions or external events. It is rather indirectly through changes in the actor's internal (mental) state. The suggested attributes are graded fairly coarse, 0, 1, 2 and 3. For instance, in the case of "discontent", this could be interpreted as appalling, bad, hopeful, and respectively good. However, here long-term goals and short-term goals have to be considered. And in the case of "relationships" a similar interpretation would be that another actor is perceived as hostile, extraneous, temporarily on the same line as oneself, or ally. How a suitable action repertoire should look like is dependent on the scenario at hand.

C. *Actor agenda*

As described in the previous section each actor has an action repertoire, and an "agenda". This repertoire contains a set of actions that can be performed and each action is associated with a probability. The reason we model each actor this way is because we want to create dynamic actors with different behaviors depending on the situation.

Each actor has relations to other actors, *e.g.*, all UN nations have some kind of relationship with the other nations. An actor also has a so called "desired state" for itself and for the other actors in the relationship. The desired state is basically desired parameter values which the actor wishes to reach, and they are used to measure distances from the actual parameter values which are simulated. The distance reflects how far away an actor is from the desired state if a specific action is performed. The longer the measured distances are for an action, the farther the actor is from the desired state and the less probable it is for an actor to choose that action. Table I shows an example of the current parameter state p_{jm}, and the desired parameter state ω_{jm}. Observe that the ω_{jm} values are viewed from a specific actor's point of view, in this case "Actor 1" which has relations to "Actor 2" and "Actor 3". If we change the point of view, then ω_{jm} is changed.

TABLE I. EXAMPLE OF ACTOR PARAMETER STATES, AND DESIRED VALUES, FROM ACTOR 1'S POINT OF VIEW.

Actor 1		**Actor 2**		**Actor 3**	
$\omega_{11}=3$	$p_{11}=2$	$\omega_{21}=0$	$p_{21}=2$	$\omega_{31}=0$	$p_{31}=2$
$\omega_{12}=2$	$p_{12}=1$	$\omega_{22}=0$	$p_{22}=3$	$\omega_{32}=1$	$p_{32}=2$
$\omega_{13}=3$	$p_{13}=2$	$\omega_{23}=0$	$p_{23}=2$	$\omega_{33}=0$	$p_{33}=2$
$\omega_{14}=3$	$p_{14}=2$	$\omega_{24}=1$	$p_{24}=1$	$\omega_{34}=0$	$p_{34}=3$
$\omega_{15}=2$	$p_{15}=2$	$\omega_{25}=0$	$p_{25}=2$	$\omega_{35}=0$	$p_{35}=1$

If an actor event is executed there will be a global parameter change, since an event has consequences. Therefore, for each action that an actor can perform, there will be parameter changes for all related actors. In this example, if Actor 1 performs action a_{11}, a_{12} and a_{13}, then there will be changes Δp_{ijkm}, on its parameters and also upon other actors' parameters, where Δp_{ijkm} is a change on parameter p_{jm}. Table II shows an example of this for Actor 1.

TABLE II. PARAMETER CHANGES FOR ACTOR 1 IF a_{11}, a_{12} AND a_{13} ARE PERFORMED BY ACTOR 1.

a_{11}	a_{12}	a_{13}
$\Delta p_{1211}=2$	$\Delta p_{1121}=1$	$\Delta p_{1131}=-3$
$\Delta p_{1311}=3$	$\Delta p_{1321}=-2$	$\Delta p_{1231}=-1$
$\Delta p_{1411}=1$	$\Delta p_{1521}=-3$	$\Delta p_{1531}=2$

For each actor relation that Actor 1 has, similar tables with actions a_{11}, a_{12} and a_{13} will be constructed. The changes Δp_{ijkm} must obviously be different for the other actors. From those table data we can now calculate parameter distances, but first we need to calculate the new parameter values after a change. We can use

$$p^*_{ijkm} = p_{jm} + \Delta p_{ijkm}, \tag{1}$$

Where p_{jm} is the old parameter value and p^*_{ijkm} the new value for actor i, parameter j, action k, and receiving actor *m*. If there are no changes for certain parameters, then we set those Δp_{kij} to 0. Now that we have the new parameter values, we can calculate the distances by

$$l_{ikm} = \sum_{j=1}^{n} | p^*_{ijkm} - \omega_{jm} | \cdot v_j, \tag{2}$$

Where v_j is a weight that determines how important the parameter is for a certain type of action. The weight v_j can, *e.g.*, be set to a number between 1 and 5. After some calculations we obtain distances, in this case table III shows an example of calculations based on the table data that we discussed earlier, where v_j is considered to be 1 for all parameters.

TABLE III. SUM OF LENGTH FOR DIFFERENT ACTIVITIES AND ACTORS.

Action	Actor 1	Actor 2	Actor 3
a_{i1}	$l_{i11} = 2$	$l_{i21} = 4$	$l_{i31} = 8$
a_{i2}	$l_{i12} = 7$	$l_{i22} = 9$	$l_{i32} = 11$
a_{i3}	$l_{i13} = 8$	$l_{i23} = 7$	$l_{i33} = 11$

From table III we can calculate the sum of distances that belong to an action that Actor 1 can perform. Once again we must observe that all values have initially been calculated from Actor 1's perspective. We can use the following formula for the aggregated sum

$$L_{a_{1k}} = \sum_{i=1}^{n} l_{1km} , \qquad (3)$$

where $L_{a_{1k}}$ is a partial sum which is based on the fact that a_{1k} is performed. For this example, this will give us the following values: $L_{a_{11}} = 14$, $L_{a_{12}} = 27$, $L_{a_{13}} = 26$. Since our model is based on the fact that the shortest distance should be the most probable (alternatively the longest distance should be least probable) we need to perform a transformation. In this case $L_{a_{11}}$ should be most probable since it is the shortest distance. We apply the inverse transform $T_{a_{1k}} = L_{a_{1k}}^{-1}$ which will give us these values: $T_{a_{11}} \approx 0.07$, $T_{a_{12}} \approx 0.04$, $T_{a_{13}} \approx 0.04$. Note that $L_{a_{1k}}$ must be non-zero. However, if a distance is 0, we can assign a low value to it, *e.g.*, 0.001. Now that we have transformed our values we can simply calculate the probabilities by standard probability methods. First, we create the sum

$$S_i = \sum_{k=1}^{n} T_{a_{ik}} , \qquad (4)$$

and then we calculate the probabilities

$$p(a_{11}) = \frac{T_{a_{11}}}{S_1} = \frac{0.07}{0.15} \approx 0.47 , \qquad (5)$$

$$p(a_{12}) = \frac{T_{a_{12}}}{S_1} = \frac{0.04}{0.15} \approx 0.27 , \qquad (6)$$

$$p(a_{13}) = \frac{T_{a_{13}}}{S_1} = \frac{0.04}{0.15} \approx 0.27 . \qquad (7)$$

Hence, the above probabilities are the new updated action repertoire for Actor 1.

The reason we used this approach for modeling the actor agenda is because we want to have a model that allows an expert (*e.g.*, a military expert) to input his knowledge into the system. From this system we could calculate the parameter changes Δp_{ijkm} for each action which increases traceability. Different factors are taken into consideration when the changes are calculated. These factors are, *e.g.*, the type of activity which the action is being carried out, in relationship to the surrounding world, amount of available resources, target types, and the actor type who is performing the action. A model which takes an expert's knowledge into count is closer to reality than models which do not.

D. *Environment objects*

As described in the previous section our model consists of actors which are groups of people that can be understood in terms of sociological models. These actors do not exist in a vacuum, but in an environment with passive objects, but yet with symbolic or functional values. These objects may consist of:

- functional buildings, such as hospitals, schools, housing, and management centers, etc.,
- transportation routes and transfer points, such as roads, bridges, pipelines, ports, airports, etc.,
- utilities such as natural resources like arable land, mines, etc., and processing facilities such as power plants, factories, warehouses, etc.,
- information channels such as radio and TV stations, networks, transmission masts, etc.,
- the symbolical sites can be geographical areas, statues or other memorials, religious buildings, etc.

It is logical to assume that environmental objects have different significance and value to different actors. Moreover, they have various levels of vulnerabilities. The impairment is graded on the scale 0, 1, 2 and 3.

E. *Tool for modelling actors and activities*

For the modeling purpose we have developed a tool called "Effect Based Planner". The tool allows the user to create and manipulate different world entities. The world entities are the actors and activities which can be assembled together in different composites.

The user starts with a predefined template which contains entities that are relevant for a specific scenario domain. From the template the user creates instances and through them the user can specify specific names and parameter values, see Fig. 3. The instances represent real actors and activities and they are used in the “scenario graph”, see Fig. 4.

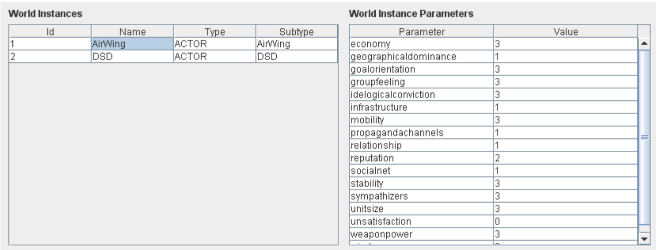

Figure 3. Instances and activities.

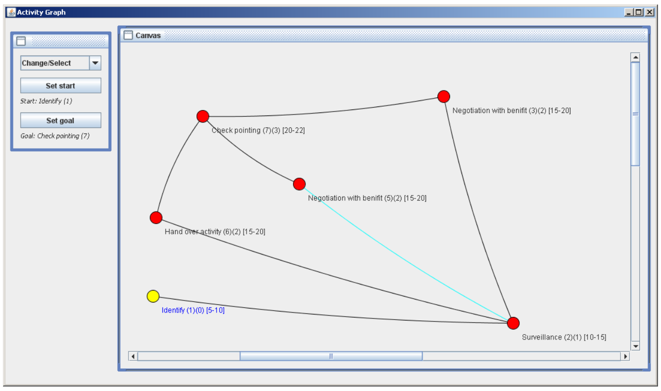

Figure 4. Scenario graph editor.

The scenario graph represents the actual plan that a decision maker has developed. Each node in the scenario graph represents an activity and each activity contains a set of actors which are grouped into different colors (subsets). The subset information is used by the event algorithm (*e.g.*, an activity event or an actor action event), in order to distinguish different actor roles in an event.

The tool also enables a user to input “wanted effect”, *i.e.*, the desired end state value for each parameter and for each actor. When a scenario graph has been built a user must also specify a start and a goal node, which set the direction of the graph traversing algorithm (A*-search).

At this stage we do not consider validation of models as we work with a fictitious scenario. With genuine models this may be done using historical scenarios with known outcomes.

VI. SIMULATION

The scenario consists of participating actors, their initial states and probability distribution for different actions, environmental data, as well as the plan that is to be evaluated. Furthermore the scenario contains an event list which consists of actions derived from the other actors’ agendas, and spontaneous/natural events. The list is dynamic and changes during the course of the simulation.

Let’s define the system state, S_n as the combination of all actors’ state parameters and all environment parameters. Consider activity A_n. It transforms system state S_n according to $S_n = f(S_{n-1}, A_n)$, in the time interval $[t_{n-1}, t_n]$. The implementation of A_n is rarely instantaneous. Instead, it is an interaction between our own activity, other actors’ agendas and response operations, and other external events, which is rather complicated. Hence, our function $f(S_{n-1}, A_n)$ is designed as an event-driven simulation model in order to manage the complex interactions in a transparent manner. The events in this case are: launching of activities (our own or any other actors’), an actor’s observations of initiated activities, and occurrence of an external event.

Furthermore the outcome of A_n can vary depending on the circumstances (the operation may even fail), which can be addressed by making the simulation stochastic, where the outcome of an activity depends on a number of random variables drawn according to some given distributions. The disadvantage of this is that we can obtain a per se reasonable, but rather unlikely outcome, which would mean that we might needlessly throw out a mostly good plan. In order to avoid this outcome we use Monte Carlo simulations, thereby obtaining a frequency function of the entire outcome space.

A consequence of implementing the function $f(S, A)$ as an event-driven stochastic simulation model is that, although the state parameters from the beginning are absolute values, after a completed action they will be represented by statistical distributions. Hence, we can choose to represent the initial states by statistical distributions as well. Similarly, the external events can be listed with typical probabilities for the actual operational theatre, season, etc.

We know that the goal of the simulation is to execute different plans and identify those plans that result in system states that are “closest” to our end state, *i.e.*, has the shortest distance. Given the approach discussed above, the *distance* to the end state will be stochastic. Hence, by calculating the distance value in each Monte Carlo loop we create the distribution of this *distance* in the form of a histogram (which approximates the frequency function). This means that the A*-algorithm (described in the next section) needs to evaluate not only a single distance value, but also the importance of the spread in the given situation. A large spread around a small average value indicates that we are on track, but this path is unstable and could easily lead to failure.

Our Monte Carlo simulation is therefore structured as follows:

```
For each round of the Monte Carlo loop:
    Initialize event list with our action A
    Randomly draw the external events and add them to the event list
    Randomly draw a starting state for each state parameter from resp. distribution
    For each actor:
          Randomly draw the next action from the current agenda and add to the event list.
    For each event in the event list as long as time is less than tn:
          Environmental parameters may change (which could generate new events).
          For each actor (including "our own" operator "):
              Note directly or indirectly through filtered or biased information
              Analyse the information → internal state and resources are changing
              Action repertoire is updated with new probabilities
              Randomly generate the next action
              Add a new action to the event list.
    Save the results for each state parameter.
Create a summary of results for each state parameter in the form of a histogram, which
serves as an approximation for resp. output distribution.
```

Figure 5. Monte Carlo algorithm.

During the actual time interval $[t_{n-1}, t_n]$ our activity A_n is initiated. Probable external events are in the same way chosen and placed in the event list according to their given distributions. The activity A_n is observed (via an information channel) by the other actors immediately or eventually. Directly, or after a period of analysis (which may be biased or colored by the information channel), the respective actor's state is changed, which can lead to a new set of probabilities in the action repertoire. An action from each actor's action repertoire is randomly chosen and placed in the event list. As the simulation proceeds and actions/events in the event list are executed new actions/events are added in the list (as the result of observations and reactions) until the end of the time interval is reached. Finally, a summary of the results for the state parameters is created. These state parameters are represented as histograms and serves as an approximation for respective output distribution.

A. A-search*

One of the main requirements of our simulation system is to be able to, at any moment in time, suggest an alternative sequence of activities that best suits the decision maker's desired end state. Such a simulation system can neither be designed according to the principle of "breadth first search" nor "depth first search". In the former case it will take too much time before we reach a reasonably correct prediction. In the latter case we get stuck with just one plan, and will not have a general view when we are asked to forecast the best approach. Instead, we apply A*-search [2, 3]. It means that, on the basis of a given system state, we simulate the effect of each alternative activity in our plan, but only one step at the time. Doing so, for every alternative, we get a new system state whose "distance" to the desired end state is calculated. Given the alternative that is best, *i.e.*, "closest" to our end state, we simulate possible subsequent alternative activities provided, but again only one step ahead in our activity/event list. One of these alternatives leads to a condition that is "closer" than the others. However, it is possible that all the alternatives actually lead away from the target as seen by Fig. 6.

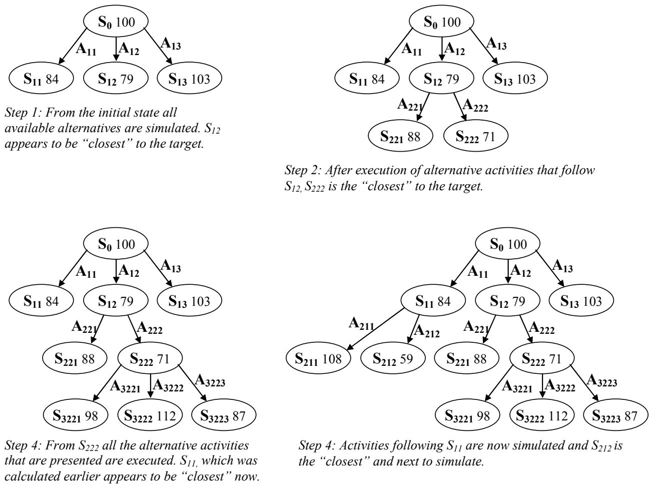

Figure 6. An example illustrating the four first steps in a simulation of a plan starting with initial system state S_0 with the *distance* of 100 to the desired end state. The available activity alternatives A_x are executed successively in the currently most favourable plan option.

Therefore, we must also compare the new "distance" with the best of the "distances" that have been simulated and recorded in the previous simulation steps, but then had opted out in favor of a better sequence of alternative activities. The best sequence now becomes the basis for the next simulation step. At any time the user can then ask for the sequence, which at that time seems to be the best, *i.e.*, the sequence of alternative activities that leads to a simulated state, which is "closest" to the desired end state. Activity lists in the investigated plans are obviously not infinite, which means that they will gradually terminate. Consequently the simulation program continues to execute the options that follow the "second best" system state. Given enough execution time all options will eventually be investigated. For the tool to function in this way the simulation system stores a list of all executed activities, the corresponding system state, and the "distance value". Therefore, the simulation kernel provides a service to store all this information in a dynamic list and is also able to restart the simulation from a previously stored state.

*B. Functions of distance calculations in A**

A central problem in applying the A*-search algorithm is to find a proper distance function. In our model the states of the actors and the environment are described by a large amount of parameters with varying resolution and weights, which complicates the task of the defining a credible distance function. The solution chosen is to define a function that calculates the distance based on the difference between parameter values of a given state and the parameter values of the end state. These differences are absolute values and are weighted according to the importance of the parameters and their impact on the success of the plan. As described earlier, our parameters are not represented as real numbers, but rather as histograms.

A state S_{i,y_i} is a vector of length n with different sub-states $S_{i,y_i,j}$, where $S_{i,y_i,j}$ is a distribution over {0, 1, 2, 3}, *e.g.*, $S_{i,y_i,j}$ = (0.2, 0.5, 0.2, 0.1) where the first 0.2 is the frequency of "0", and 0.5 the frequency of "1", etc. We have

$$S_{i,y_i} = (S_{i,y_i,1},\ S_{i,y_i,2},\ \ldots,\ S_{i,y_i,n}), \tag{8}$$

where y_i is the current sequence of choices made for all activities A_1 to A_i. The initial stated is called $S_{0,0}$, and the end state is called S_e.

The distance $\Delta(S_{i,y_i}, S_{i+1,y_{i+1}})$ between two successive states S_{i,y_i} and $S_{i+1,y_{i+1}}$ is calculated as

$$\Delta\left(S_{i,yi}, S_{i+1,y_{i+1}}\right)= \frac{1}{\mu[A_{i+1,y_{i+1}(y_{i+1})}]\,\omega[A_{i+1,y_{i+1}(y_{i+1})}]}\sum_{j=1}^{n} w_j D_S\left(S_{i,y_i,j}, S_{i+1,y_{i+1},j}\right), \tag{9}$$

where the distance D_S is calculated as

$$D_S\left(S_{i,y_i,j}, S_{i+1,y_{i+1},j}\right) = \sum_{k=0}^{3}\left|S_{i,y_i,j}(k) - S_{i+1,y_{i+1},j}(k)\right|, \quad (10)$$

and where w is a vector of length n with elements w_j, where $w_j \in [0,1]$ are weights assigned during modeling to address the relative importance between different $\left\{S_{i,y_i,j}\right\}_{j=1}^{n}$. The distance from the starting state $S_{0,0}$ to a current state S_{x,y_x}.

The ω function is *a priori* information regarding the importance of the activity from an effects-based perspective. This information is obtained from a CIM. We have

$$\omega[A_{i+1,y_{i+1}(y_{i+1})}] = \max_q\{CIM[A_{i+1,y_{i+1}(y_{i+1})}, SE_q]\}. \quad (11)$$

The μ function records the decision maker's current interest in a particular activity (see Sec. IV). We have

$$\mu[A_{i+1,y_{i+1}(y_{i+1})}] = \prod_j \mu_j\left[A_{i+j,y_{i+1}(y_{i+1})}\right], \quad (12)$$

where $\{\mu_j\}_j$ are drawn from the views that decision makers use to give priorities of simulations tasks to the simulator.

The distance from the initial state $S_{0,0}$ to a current state S_{x,y_x} is given by

$$g(y_x) = \sum_{i=0}^{x-1}\Delta\left(S_{i,y_i}, S_{i+1,y_{i+1}}\right). \quad (13)$$

The estimated distance from the current state to the end state is given by

$$h(y_x) = \Delta\left(S_{x,y_x}, S_e\right). \quad (14)$$

With the total distance from the initial state to the end state via the current state is

$$f(y_x) = g(y_x) + h(y_x). \quad (15)$$

This is the distance function that is minimized by A*.

C. *The simulation tool*

The Effect Based Planner contains a simulation section where the scenario can be simulated, see Fig. 7. The user can specify simulation delay, the number of actor reactions and the detail level of debug information. The user can also pause and stop the simulation.

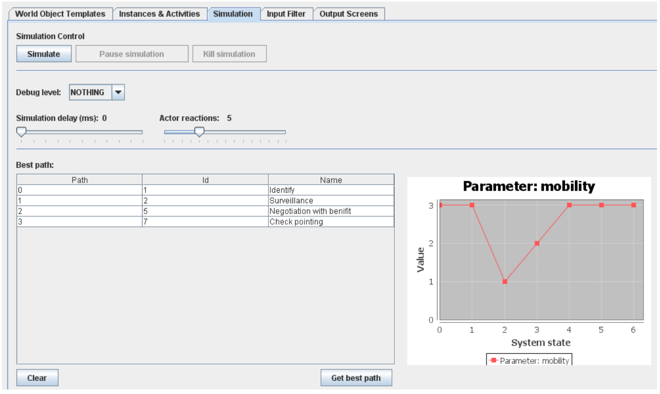

Figure 7. Simulation section.

The simulation section shows two important results in real-time. The first result is a list that shows the current optimal path which is described by a chain of activities. If the activities in the list are performed with these alternatives, the distance from the current state to the end state is minimized. The second result is a diagram that shows the optimal individual parameter development during system state changes for each actor. Using this information the decision maker will know how each parameter has changed historically during the simulation, which will increase traceability.

The simulation model is stored using an XML-based format with a schema. A user can easily change the plan model, perform a simulation, gain new results, update the plan and store it once again so that it can be further used in the future.

Plan simulation is performed by the simulation engine. The engine basically contains an implementation of the A*-search algorithm which uses the Monte Carlo principle for event based simulation. During simulation, each system state in the A*-search algorithm is stored. A system state is a snapshot of the parameter state for all entities at a specific time during the execution of the algorithm. This state contains crucial information which is used by the simulation control (input filter) in order to narrow the A*-search.

The internal data that is used by the simulation is stored in a separate container. This way it can easily be serialized and used by other simulation engine instances.

The simulation engine has an optimization parameter which is regulated by the user. This parameter determines how long the event reaction chain can be for each actor. By default each actor can react once to a certain event, but this global value can be changed in order to gain more realistic results. For example if an actor i can perform a_{i1}, a_{i2} and a_{i3}, and the optimization parameter is set to 1, then he may only choose one of the actions as a response to an event, that is, he may only react once. However, we can specify this parameter to be larger.

VII. DECISION SUPPORT

Decision support is given as a set of plans that are similar in structure and consequences. That they are similar in structure means that they have more or less performed similar alternative activities. Similar in consequences means that they travel on average the same distance towards the end state for each performed activity. These plans are robust as there are always several alternative plans that can be used if the current plan must be abandoned. Dynamic replanning can be performed as a selection of one of the similar plans within this set of plans.

During simulation an assessment is made of how well each activity is performed. All such estimates are based on various simulation tasks and stored in order to be rapidly re-used by future simulation and is also transferred to the decision support system so that a consolidated assessment can be made. The compilation of partial results from all simulation activities is done by the g-function, Eq. (13), that measures the consequence of all performed activities as a distance from the initial state to the current simulated state.

We observe the difference in consequences between two plans. We compare the incremental changes of g called Δg as each plan P_i and P_j progresses down the sequence of additional activities A_k.

In addition, we need to measure the structural distance between two plans. This is done by the Hamming distance H [15] which measures the structural distance between P_i and P_j. We have,

$$H(P_i.A_k, P_j.A_k) = \begin{cases} 0, & P_i.A_k = P_j.A_k \\ 1, & P_i.A_k \neq P_j.A_k \end{cases} \qquad (16)$$

when both activities $P_i.A_k$ and $P_j.A_k$ exists within the simulated sequences P_i and P_j, otherwise 0 by definition.

Using this measure, we compare each activity in two different plans to calculate the structural distance between the plans. For each activity we observe the alternative chosen in both plans.

We need to find plans that are close in both structure and consequence so that one can work as an alternative to the other should dynamic replanning be necessary. We put these two measures together into an interaction functions J_{ij}^- that measures the overall distance between plan P_i and P_j.

We have,

$$J_{ij}^- = 1 - \left[1 - \sum_k H(P_i.A_k, P_j.A_k) \right] \times \left[1 - \sum_k \left| \Delta g(P_i.A_k) - \Delta g(P_j.A_k) \right| \right] \qquad (17)$$

where the incremental change of g for P_i between activities A_{k-1} and A_k is

$$\Delta g(P_i.A_k) = g(P_i.A_k) - g(P_i.A_{k-1}), \qquad (18)$$

and identically for P_j.

We partition the set of all simulated plans into clusters in such a way as to minimize the overall sum of all interactions J_{ij}^- within each cluster, Fig. 8.

INITIALIZE

K (number of clusters); N (number of plans);

Assign $J_{ij}^- \ \forall i, j$; $s = 0$; $t = 0$; $\varepsilon = 0.001$; $\tau = 0.9$; $\gamma = 0.5$;

$T_0 = T_c$ (a critical temperature) = $\frac{1}{K} \cdot \max(-\lambda_{\min}, \lambda_{\max})$,

where $\lambda_{\min}$ and $\lambda_{\max}$ are the extreme eigenvalues of M, where $M_{ij} = J_{ij}^- - \gamma\delta_{ij}$;

$V_{ia}^0 = \frac{1}{K} + \varepsilon \cdot rand[0,1] \ \forall i, a$;

REPEAT

• REPEAT-2

$\forall i$ Do:

$$H_{ia}^s = \sum_{j=1}^{N} J_{ij}^- V_{ja}^{\begin{cases} s+1, j<i \\ s, j \geq i \end{cases}} - \gamma V_{ia}^s \ \forall a ;$$

$$F_i^s = \sum_{a=1}^{K} e^{-H_{ia}^s / T^t} ;$$

$$V_{ia}^{s+1} = \frac{e^{-H_{ia}^s / T^t}}{F_i^s} + rand[0,1] \ \forall a ;$$

$s = s + 1$;

UNTIL-2

$$\frac{1}{N} \sum_{i,a} \left| V_{ia}^s - V_{ia}^{s-1} \right| \leq 0.01 ;$$

• $T^{t+1} = \tau \cdot T^t$;

• $t = t + 1$;

UNTIL

$$\frac{1}{N} \sum_{i,a} \left(V_{ia}^s \right)^2 \geq 0.99 ;$$

RETURN

$$\left\{ \chi_a \middle| \forall S_i \in \chi_a . \ \forall b \neq a \ V_{ia}^s > V_{ib}^s \right\};$$

Figure 8. Potts spin clustering of simulated plans partition the set of simulated plans into clusters of similar plans.

To find the number of clusters K we plot the energy function

$$E = \frac{1}{2}\sum_{a=1}^{K}\sum_{i,j=1}^{N} J_{ij}^{-} V_{ia} V_{ja} \tag{19}$$

in a graph for different number of clusters K. We use a convex hull algorithm to calculate the lower envelope of E. At an arbitrary abscissa, the envelop function is bisected in a left and right part, each of which is fitted by least squares to a straight line. The acute angle between the two lines is maximized over all bisection abscissas and the maximizing abscissa is chosen as the number of clusters [16, p. 90].

Potts spin clustering is an effective method and as such does not guarantee an optimal solution. However, this is not considered a problem as our focus is on providing a set of robust alternative plans should replanning be necessary, and not one single second best plan.

These clusters are sets of alternative plans available, should replanning be necessary. If a plan is in the midst of execution the decision maker can observe evaluations of alternative continuations of the plan, and see which alternative activities to avoid and which are preferable as they are within a robust subset of plans, Fig. 9.

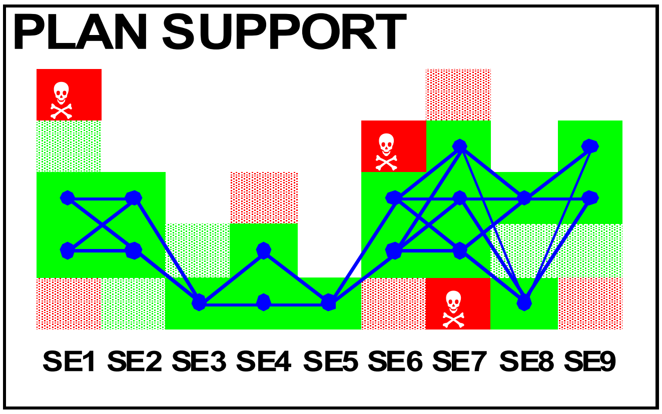

Figure 9. Plan support is given a set of robust plans illustrated by green alternastives. Each column represent an activity (*e.g.*, supporting effect) of the plan. The rows are different alternatives for these activities.

Plans are judged by their robustness. This is measured, not by the score the plan receives itself, but rather by the minimum score of all other plans that are close in structure and in their consequences. When executing a plan like this, we have a robust situation where there are similar plans with minor differences in both structure and consequence. They function as alternatives if dynamic replanning becomes necessary.

VIII. CONCLUSIONS

We have developed a simulation-based decision support methodology with which we can test operational plans as to their robustness. Primarily, this methodology highlights the dangerous options in an operational plan, leaving the decision maker free to focus his attention on the set of remaining robust plans.

REFERENCES

[1] F. Moradi and J. Schubert, "Modelling a simulation-based decision support system for effects-based planning," in Proceedings of the NATO Symposium on Use of M&S in: Support to Operations, Irregular Warfare, Defence Against Terrorism and Coalition Tactical Force Integration (MSG-069), paper 11, pp. 1−14, October 2009.

[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, pp. 269–271, December 1959.

[3] P. E. Hart, N. J. Nilsson, B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, pp. 100–107, July 1968.

[4] F. Y. Wu, "The Potts model," Reviews of Modern Physics, vol. 54, pp. 235–268, January 1982.

[5] J. Schubert, "Managing inconsistent intelligence," in Proceedings of the Third International Conference on Information Fusion, Paris, France, 2000, vol. 1, pp. TuB4/10–16, July 2000.

[6] J. Schubert, "Clustering belief functions based on attracting and conflicting metalevel evidence using Potts spin mean field theory," Information Fusion, vol. 5, pp. 309–318, December 2004.

[7] E. A. Smith, Complexity, Networking, and Effects-based Approaches to Operations. Washington, DC: Department of Defense CCRP, 2006.

[8] J. P. Hunerwadel, "The effects-based approach to operations: Questions and answers," Air & Space Power Journal, vol. 20, pp. 53−62, Spring 2006.

[9] Effects-based Approach to Multinational Operations, Concept of Operations (CONOPS) with Implementation Procedures, Version 1.0. Suffolk, VA: United States Joint Forces Command, 31 July 2006.

[10] E. A. Smith, Effects-Based Operations, Applying Network Centric Warfare in Peace, Crisis and War. Washington, DC: Department of Defense CCRP, 2002.

[11] P. S. E. Farrell, "New operations decision support requirements derived from a control theory model of effects-based thinking," in Proceedings of the 13th International Command and Control Research and Technology Symposium, paper 248, pp. 1−17, June 2008.

[12] J. Schubert, M. Wallén, and J. Walter, "Morphological refinement of effect-based planning," in Stockholm Contributions to Military-Technology 2007, M. Norsell, Ed. Stockholm: Swedish National Defence College, 2008, pp. 207−220.

[13] J. Schubert, "Analysis and assessment of effects-based plans," in Proceedings of the NATO Symposium on Analytical Support to Defence Transformation (SAS-081), paper 33, April 2010.

[14] E. Hudlicka, G. Zacharias, W. Rouse, and K. Boff, Eds., Requirements and Approaches for Modeling Individuals within Organizational Simulations. Hoboken, NJ: Wiley, 2005.

[15] R. W. Hamming, "Error detecting and error correcting codes," Bell System Technical Journal, vol. 29, pp. 147–160, April 1950.

[16] S. Ahlberg, P. Hörling, K. Johansson, K. Jöred, H. Kjellström, C. Mårtenson, G. Neider, J. Schubert, P. Svenson, P., Svensson, and J. Walter, "An information fusion demonstrator for tactical intelligence processing in network-based defense," Information Fusion, vol. 8, pp. 84−107, January 2007.