Modelling a Simulation-Based Decision Support System for Effects-Based Planning

Dr. Farshad Moradi and Dr. Johan Schubert
Division of Information Systems
Swedish Defence Research Agency
SE-164 90 Stockholm
Sweden

Email: farshad.moradi@foi.se, johan.schubert@foi.se

ABSTRACT

Models constitute an important component in decision support functions of C4I systems. They improve the military commander's abilities to create situation awareness, analyse threats and make proper decisions. Computerized models can be exploited for the purpose of simulation, which enables us to cover a much wider range of options and go deeper in impact assessments. In this paper we describe decision support and simulation techniques to facilitate Effects-Based Planning (EBP). In our approach, 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 (EBAO) 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. The simulation of plans is designed to deliver results, indicating the (so far) best sequence, at each point of time. Hence, we have chosen to use the A-search algorithm for traversing through the activity tree and choosing the next activity to be simulated. This method helps us to decide at any point of time which sequence of activities has the best result so far, i.e., has resulted in a system state that is "closest" to our end-state. The paper also includes a description of our model, different objects, their relations, and the structure of our simulation kernel. The system is still under development hence there are no experimental results obtained so far.*

1.0 INTRODUCTION

Models constitute an important component in decision support functions of C4I systems. The work of military commanders was previously mainly based on using mental models and maps for creating situation awareness, analysing threats, investigating different actions and making decision based on those. The development of computers has meant opportunities for a radical increase in capacity for these activities. By leveraging databases with information about the enemy, environment and past experience, we are strengthening our models and thereby our understanding. By further formulating our models in computerized form we also increase our ability to exploit them for simulations, by which we can both cover a much wider range of options and go deeper in impact assessments [1].

With regard to information interpretation, i.e., creation of situation awareness and situation understanding and threat analysis, models play a central role when it comes to assimilate large amounts of complex information from various sensors (including humans). One application is Information Fusion, which can be seen as a real-time simulation of the real course of events. Information Fusion gives the commander and his staff better ability to quickly discern which reports are linked and what is going on. It also improves the commander’s ability to control its sensory resources to achieve desired situational awareness [2].

Another way to use models in information interpretation is the so-called indicator modelling [3]. Here we assume a *case database* where situations from previous missions are stored together with a set of events that follow that situation, and a final event, such as a riot. With an appropriate situation model, we can then compare a current situation with the database situations, and select those that are most similar. The decision-maker is thus made aware of the similar corollary events, which indicate that the corresponding end event can occur even now.

As for evaluating the impact of alternative decisions, it is possible to use expert systems - models which simulate experts’ way of drawing conclusions [4]. They allow for “faster” decision-making for the experienced, and “safer” decisions for the less experienced, by ensuring that “no” relevant aspects are forgotten. The disadvantage is that they can easily lead to a stereotyped and predictable behaviour, which might be partly handled by making the system self-learning.

However, a more flexible decision support can be achieved with a tool that in addition to an expert system also includes a model with which we can simulate the events and produce outcomes resulting from making alternative decisions [5]. Such a tool should not only simulate a single thread of activities, but rather calculate statistical values of outcomes of different threads. It must also have a high credibility, and be fast to allow many simulations. Hence, the model may not be too detailed. Nevertheless, large plans with many alternative activities may take a long time to execute. A more practical simulation system should be able to, at any moment in time, suggest an alternative that best suits the commander’s criteria of a successful plan. Furthermore, it must have a flexible and easy to use interface, which enables us to quickly and easily define different alternatives and modify the model as our knowledge of other actors’ characteristics/capabilities grows.

“Real-time Simulation for Supporting Effects-Based Planning” is an ongoing research project at the Swedish Defence Research Agency, which was initiated in 2008 with the goal of designing and developing a simulation-based decision support system for supporting the planning process of the Effects-Based Approach to Operations (EBAO) [6, 7]. In this paper we present the approach employed in the project to achieve the above goal. The decision support system enables a decision maker to test a number of feasible plans against possible courses of events and decide which of these plans is capable of achieving the desired military end-state. The purpose is to evaluate plans and understand their consequence through simulating the events and producing outcomes which result from making alternative decisions. The simulation of plans with a sequence of alternative activities is designed to deliver results, indicating the (so far) best sequence, at each point of time. Hence, we have chosen to use the A*-search algorithm [8, 9] for traversing through the activity tree and choosing the next activity to be simulated. This method helps us to decide at each point of time which sequence of activities has the best result so far, i.e., has resulted in a system state that is “closest” to our end-state.

The remainder of the paper is structured as follows. In Sec. 2 we describe the overall Decision Support System giving support on operational planning by testing and evaluating alternative operational plans. In Sec. 3 we present our model, different concepts, and their relations. Sec. 4 gives an introduction to our simulation and its structure. Finally, Sec. 5 concludes the paper with current status of the project, general conclusions and future work.

2.0 DECISION SUPPORT

In EBAO as in any other planning process, there is a need to assess possible plans before execution (Effects-Based Execution (EBE) in the context of EBAO) and to perform re-planning when necessary. The aim is to evaluate the plans, including discovering its weaknesses and understanding its implications. An important prerequisite for good planning is to find and use appropriate indicators so that intelligence questions can be asked.

2.1. Analyzing the operational plan

Analyzing and simulating the operation plan can be made at any time. When an operation begins to take shape one should be able to analyze and simulate several alternative plans that are in the main direction of interest. This task uses large numbers of simulations with different alternative plans against various possible scenarios. The goal is to find robust groups of plans that have similar implications.

We assume the plans and let them control the evaluation process. All activities of the plans are to be simulated against all events in the chain of events. The operational plan is simulated by providing a wide range of simulation tasks to the simulator. Any such assignment is made up of a particular alternative for a particular activity. This simulation task has a specific location in a tree of plans, where each level represents a new activity, and each branch one of the different options available for this activity.

The decision maker can prioritize between different plans by letting the simulator know his current area of interest, times of interest, and through an activity clustering choose groups of prioritized activities, see figure 1.

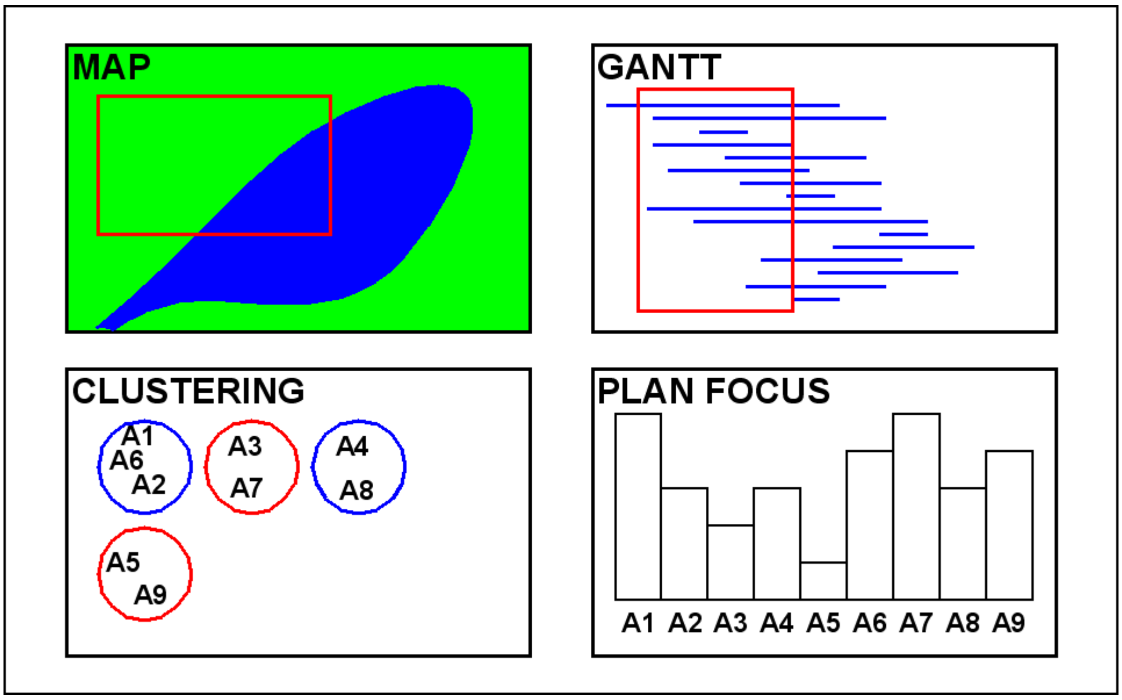


Figure 1: Through an input interface the user may select which part of the plan the simulator should focus on such as, an area in a map, start and end time in a Gantt schema, or clusters where activities that strongly influence each other are grouped together. These inputs are fused in the lower right area using the function μ (see below).

This guidance will allow the simulator to focus on alternatives that are of interest to the decision maker. The function μ is showing decision-maker current interest in a particular activity. We have,

$$\mu[A_{i+1,\, y_{i+1}(y_{i+1})}] = \prod_j \mu_j[A_{i+1,\, y_{i+1}(y_{i+1})}]$$

where $\{\mu_j\}$ is drawn from all the views that the decision-maker uses for his prioritization of simulation tasks to the simulator.

An event's overall significance from an effects-based approach is obtained by the *a priori* information given by the function ω. This information is retrieved from a cross-impact matrix (CIM) [10]. We have,

$$\omega[A_{i+1,\, y_{i+1}(y_{i+1})}] = \max_q \left\{ \text{CIM}\,[A_{i+1,\, y_{i+1}(y_{i+1})}, SE_q] \right\}$$

An assessment is made of how well each activity is managed by its mission. All such estimates based on various simulation tasks are stored in order to rapidly be re-used by future simulation and is transferred to decision support system so that a consolidated assessment can be made. The compilation of partial results from all simulation activities can be made by the statistical method developed for subjective Effects-Based Assessment (EBA) [11], where each assessment activity takes place separately, and are then fused statically to an overall assessment of entire operation plan. 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 structurally close, as well close in their consequences.

We use an information measure to measure the structural distance between two plans P_i and P_j, or between two events H_i and H_j. We choose the Hamming distance [12]

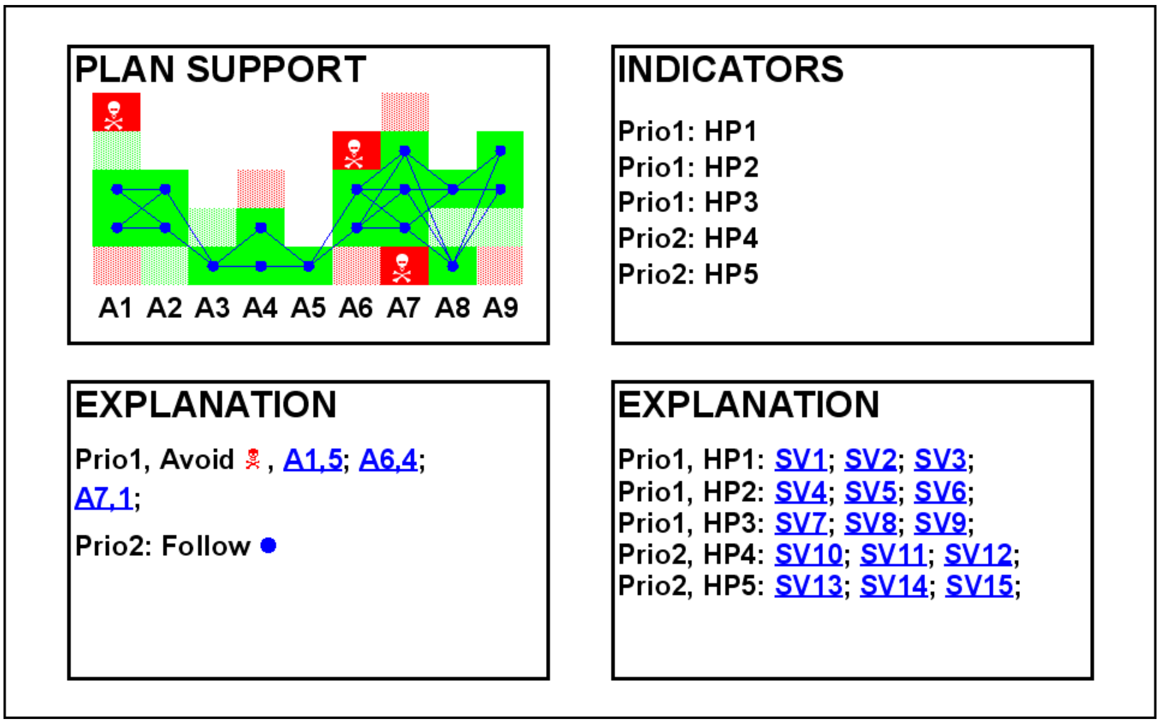
$$\text{Hamming\_distance}(P_i, P_j) = \sum_k \begin{cases} 0, & P_i.A_k = P_j.A_k \\ 1, & P_i.A_k \neq P_j.A_k \end{cases}$$

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

2.2. Finding indicators

We can support the intelligence service by finding key indicators through a range of simulations. The hypothesis is that there are families of plans with family of events that have similar consequences.

We seek to find indicators that intelligence at a later stage can use to determine if a turn of events belongs to one or other family of events. These indicators describe the dividing line between groups of different plans with similar consequences. The idea is that for all events in the group, all have similar consequences.


PLAN SUPPORT

A1 A2 A3 A4 A5 A6 A7 A8 A9

INDICATORS

Prio1: HP1
Prio1: HP2
Prio1: HP3
Prio2: HP4
Prio2: HP5

EXPLANATION

Prio1, Avoid ☠ , A1,5; A6,4; A7,1;

Prio2: Follow ●

EXPLANATION

Prio1, HP1: SV1; SV2; SV3;
Prio1, HP2: SV4; SV5; SV6;
Prio1, HP3: SV7; SV8; SV9;
Prio2, HP4: SV10; SV11; SV12;
Prio2, HP5: SV13; SV14; SV15;

Figure 2: In the output interface, decision support is given regarding the most robust operational plans, with explanations on potential problems to avoid (left side). A second output is on indicators found by the systems. These are events boundaries between events that lead to drastic consequences if plans cross them (right side).

We use the Hamming distance to measure the structural distance between the plans and add to this an absolute distance of its consequences.

Finally, all plans are clustered with their consequences [13]. The difference of the event parts in between clusters are the indicators (see figure 2, right side). They can be found using a support vector machine [14, 15].

3.0 MODELLING ACTORS AND ACTIVITIES

In order to develop the simulation system which is the heart of our decision support tool, two main questions have to be answered;

- How should the reality be modelled and what aspects to be captured?
- How should the simulation engine be designed?

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

3.1 Plan and activities

A *plan* as it has been defined in the context of EBAO is a sequence of *activities* that together lead to a desired *end-state* which is set by a military force. These activities, which can be considered as events initiated by own forces, require different types of resources in order to be executed. They can affect each other and be affected by external events. These external events can either be initiated by other actors or be spontaneous/natural events. The former could be planned, i.e., an actor's action according to its agenda and 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 unpredicted incidents, such as weather conditions, natural catastrophes, an unprovoked attack or an accident.

Based on the above discussion there are three different types of events in our model: the launch of an activity (our own action or any other actor's action) observations and reactions made by an actor, external events. Each event is interpreted by each actor as more or less hostile or friendly. It depends on the state of the actors and their relations, such as degree of aversion. It 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. 3.3), e.g., lowers their functionality with 2 units.

3.2 Actors

An *actor* is defined 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 [16]. They may be more or less clearly defined and organized, everything from police forces, relief agencies, well-organized militia units, and state administrative bodies to loosely coupled groups and social clusters, which are only 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, a political leaders or a financial potentate. A special actor is "we", i.e., the group that is to use this simulation.

The *action repertoire* is a set of possible actions that an entity is capable of performing. It is determined by its resources and knowledge, and what 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 for instance 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. 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 the activities and events, together with the probability of performing different actions. Hence, each action has a probability associated with, which are changed according to some functions. Actor attributes can be seen in table 1.

Table 1: Actor attributes divided into resources and internal state

Resources:	Weapon Strength: firepower, movement
	Crew: number capable of bearing arms, no of sympathizers, location
	Economy : scale, stability, spatial dominance
	Logistical capacity to use resources optimally: infrastructure, propaganda channels
	Soft power: contacts, reputation
Internal state:	discontent - perceived distance to the ideal desired end-state
	relationships - the degree of aversion to each of the other players
	teamwork - cohesion
	ideological conviction
	purposefulness
	cunning - wisdom

Probabilities of performing actions and the agenda are not directly affected by the other actors' actions or external events. It is rather 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 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 obviously dependent on the scenario at hand. The following table presents an example of possible actions for two types of actions, military actions and guerrilla.

Table 2: Military and Guerrilla type of actions as represented in an action repertoire

Military	Guerrilla
Bomb plants	Destroy infrastructure
Bomb transports	Mass kidnapping
Insulate and tie the opponent's resources	Attack on authority outposts
Regroup	Destroy crops
Eliminate opponents positions	Hijack transports
Secure transport corridor	Destroy depots
Secure storage area	
Secure area	
Search an area	
Prevent view	
Sniper	
Capitulation	

3.3 Scenario and Environment

The simulation scenario consists of participating actors, their initial state 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.

The environment consists of various facilities and sites with symbolic value. The facilities may consist of:

- Functional buildings, such as hospitals, schools, housing, and management centres, 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.

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.

4 SIMULATION

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 perhaps passive object but yet with symbolic or functional value. The system state, S_n can therefore be described by all actors' state parameters and all environment parameters.

Now consider the activity A_n. It transforms the 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 is 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 a mostly good plan. In order to avoid this 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 will be represented by statistical distributions. Hence, we choose to represent the initial states by statistical distributions. Similarly, the external events can be listed from the beginning with typical probabilities for the actual operational theatre, season, etc.

The goal of the simulation is to execute different plans and identify those plans that result in system state that are "closest" to our end-state, i.e., has the shortest distance. Given the above approach the *distance* to the end-state will be stochastic. 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 requires the A*- algorithm (described in the next section) to evaluate not only a single distance value, but also the importance of the spread in the given situation. A large spread around a little average value indicates that we are on track, but that path is very unstable and could easily lead to failure.

During this actual time interval (t_{n-1}, t_n) our activity A_n is initiated. It 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 coloured by the information channel), 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.

Probable external events are in the same way chosen and placed in the event list according to their given distributions. 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. Our Monte Carlo simulation is therefore structured as follows:

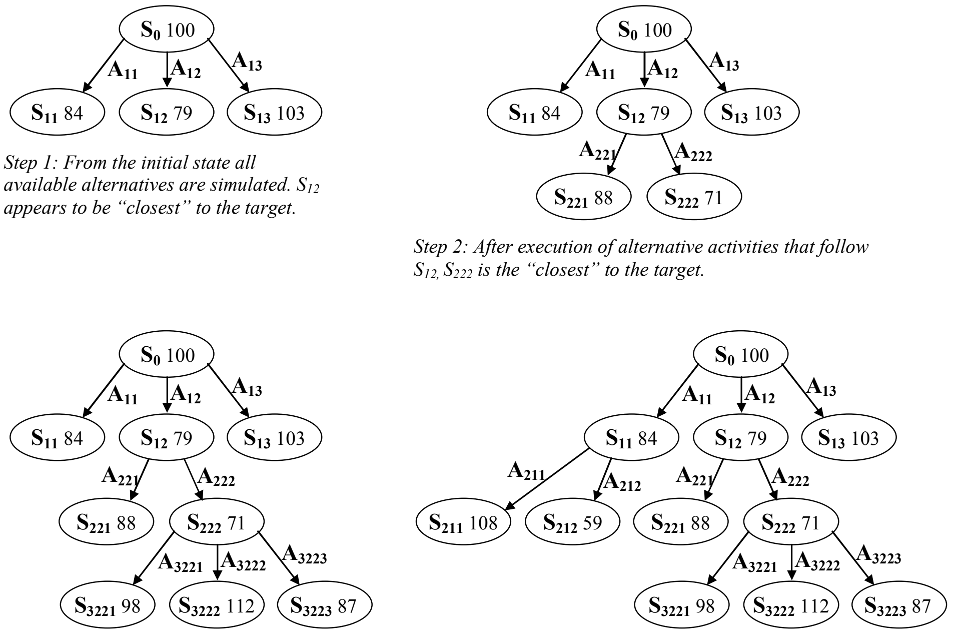
```
For each round of the Monte Carlo loop:
    Initialize event list with our action A
    Randomly draw the external events and add them to the event list
    Randomly draw a starting state for each state parameter from resp. distribution
    For each actor:
        Randomly draw the next action from 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.
```

4.1 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 probably a long time before we reach a reasonably correct prediction. In the latter case we get stuck in a subset of plan options (alternative activities), and probably will not have a general view when we are asked to forecast the best approach. Instead, we are applying the combination, known as A*-search [8, 9]. It means that we on the basis of a given system state simulate the effect of each relevant option in our plan, but only one step at the time. Doing so, for every option we get a new system state, whose "distance" to the desired end-state is then calculated. Given the option that is best, i.e., "closest" to our end-state, we simulate possible subsequent options provided, but again only one step ahead in our activity/event list. One of these options leads to a condition that is "closer" than the others. However, it is possible that all the options actually lead away from the target as seen by figure 3. 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 favour of a better option. The best option now becomes the basis for the next simulation step. At any time the user can then ask for the option, which at that time seems to be the best, i.e., the option that leads to a simulated state, which is "closest" 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". Simulation kernel provides therefore services to store all this information in a dynamic list and also be able to restart the simulation from a previously stored state.

4.1.1 Functions of distance calculations in A*

A central problem in applying 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 weight, which complicates the task of the defining a credible distance function. The solution chosen in this case 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 rather as histograms.



Step 1: From the initial state all available alternatives are simulated. S_{12} appears to be "closest" to the target.

Step 2: After execution of alternative activities that follow S_{12}, S_{222} is the "closest" to the target.

Step 4: From S_{222} all the alternative activities that are presented are executed. S_{11}, which was calculated earlier appears to be "closest" now.

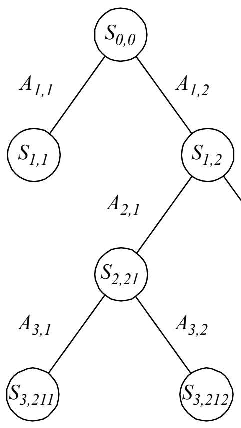
Step 4: Activities following S_{11} are now simulated and S_{212} is the "closest" and next to simulate.

Figure 3: 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.

A state $\boldsymbol{S}_{i,y_i}$ 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.,

$$\boldsymbol{S}_{i,y_i} = (S_{i,y_i,1}, S_{i,y_i,2}, \ldots, S_{i,y_i,n}),$$

where $\boldsymbol{y}_i$ is the current sequence of choices made for all activities A_1 to A_i. The initial stated is called $\boldsymbol{S}_{0,0}$, and the end-state is called $\boldsymbol{S}_e$.



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

$$\Delta(\boldsymbol{S}_{i,y_i}, \boldsymbol{S}_{i+1,y_{i+1}}) = \frac{1}{\mu[A_{i+1,\boldsymbol{y}_{i+1}(y_{i+1})}]\omega[A_{i+1,\boldsymbol{y}_{i+1}(y_{i+1})}]}\sum_{j=1}^{n} w_j D_S(S_{i,\boldsymbol{y}_i,j}, S_{i+1,\boldsymbol{y}_{i+1},j})$$

where the distance D_S is calculated as

$$D_S(S_{i,\boldsymbol{y}_i,j}, S_{i+1,\boldsymbol{y}_{i+1},j}) = \sum_{k=0}^{3}\sum_{l=0}^{3} |k-l| \cdot S_{i,\boldsymbol{y}_i,j}(k) \cdot S_{i+1,\boldsymbol{y}_{i+1},j}(l)$$

and where $\boldsymbol{w}$ 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 $\{S_{i,\boldsymbol{y}_i,j}\}_{j=1}^{n}$. The distance from the starting state $\boldsymbol{S}_{0,0}$ to a current state $\boldsymbol{S}_{x,y_x}$.

$$g(\boldsymbol{y}_x) = \sum_{i=0}^{x-1} \Delta(\boldsymbol{S}_{i,y_i}, \boldsymbol{S}_{i+1,y_{i+1}})$$

$$h(\boldsymbol{y}_x) = \Delta(\boldsymbol{S}_{x,y_x}, \boldsymbol{S}_e)$$

where the total distance is

$$f(\boldsymbol{y}_x) = g(\boldsymbol{y}_x) + h(\boldsymbol{y}_x).$$

5 CONCLUSIONS

In this paper we present our design of 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. Furthermore, we have suggested a methodology that can find important indicators, towards which the intelligence service may put intelligence questions. Results of these questions should work as early warnings if plans start to go wrong, leaving some time for re-planning. The system is still under development hence there are no experimental results obtained so far. We are currently testing the first version of the system. Our results are preliminary and no precise conclusions can be drawn about adapted design choices. Future work includes testing the system with actual operational plans, a more precise actor profiles, and detailed functions for calculating probabilities of actions in the action repertoires.

6 REFERENCES

[1] Zeigler, B.P., Praehofer, H., Kim, T.G. (2000), *Theory of Modeling and Simulation*, Second Edition. Academic Press, Inc., Orlando, FL.

[2] Hall, D.L., Llinas, J. (Eds.) (2001), *Handbook of Multisensor Data Fusion*. CRC Press, Boca Raton, FL.

[3] Kolodner, J. (1993), *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.

[4] Hayes-Roth, F. (Ed.) (1983), *Building Expert Systems*. Addison-Wesley, Reding, MA.

[5] Schubert, J., Ferrara, L., Hörling, P., Walter, J. (2008), A decision support system for crowd control, in *Proceedings of the 13th International Command and Control Research Technology Symposium*, Seattle, USA, 17–19 June 2008, pp. 1−19.

[6] Smith, E.A. (2006), *Complexity, Networking, and Effects-Based Approaches to Operations*. U.S. Department of Defense CCRP, Washington, DC.

[7] Effects-Based Approach to Multinational Operations, Concept of Operations (CONOPS) with Implementing Procedures, Version 1.0 (for comment) (2006). United States Joint Forces Command, Joint Experimentation Directorate, Suffolk, VA.

[8] Dijkstra E.W. (1959), A note on two problems in connexion with graphs, *Numerische Mathematik*, **1**(1):269–271.

[9] Hart, P.E., Nilsson, N.J, Raphael, B. (1968), A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, **4**(2):100–107.

[10] Schubert, J. (2008), Subjective Effects-Based Assessment, in *Proceedings of the Eleventh International Conference on Information Fusion*, Cologne, Germany. ISIF, Mountain View, CA, pp. 987–994.

[11] Schubert, J., Wallén, M., Walter, J. (2008), Morphological refinement of Effect-Based Planning, in M. Norsell (Ed.), *Stockholm Contributions to Military-Technology 2007*. Swedish National Defence College, Stockholm, pp. 207–220.

[12] Hamming, R.W. (1950), Error detecting and error correcting codes, *Bell System Technical Journal*, **29**(2):147–160.

[13] Schubert, J., Sidenbladh, H. (2005), Sequential clustering with particle filters - Estimating the number of clusters from data, in *Proceedings of the Eighth International Conference on Information Fusion*, Philadelphia, USA. IEEE, Piscataway, NJ, Paper A4-3, pp. 1−8.

[14] Vapnik, V., Lerner A. (1963), Pattern recognition using generalized portrait method, *Automation and Remote Control*, **24**:774–780.

[15] Boser, B.E., Guyon, I.M., Vapnik, V.N. (1992), A training algorithm for optimal margin classifiers, in D. Haussler, (Ed.), in *Proceedings of the Fifth Annual ACM Workshop on COLT*, Pittsburgh, PA. ACM Press, New York, NY, pp. 144–152.

[16] Hudlicka, E., Zacharias, G., Rouse, W., and Boff, K. (Eds.) (2005), *Requirements and Approaches for Modeling Individuals within Organizational Simulations*. John Wiley & Sons, Inc., Hoboken, NJ.