The Modeling and Analysis of Computer Generated Forces Representing Groups and Organizations in Military Conflict Zones

Linus J. Luotsinen and Eric Sjöberg
Department of Informatics and Aero Systems
Swedish Defence Research Agency
SE-164 90 Stockholm
SWEDEN

Email: linus.luotsinen@foi.se, eric.sjoberg@foi.se

ABSTRACT

Computer generated forces (CGFs) are autonomous, computer controlled, entities employed to model human actors in many simulation-based training and decision support tools. In this work we present a CGF modeling approach targeted towards modeling aggregate entities representing groups and organizations such as civilians, armed forces, etc. in military conflict zones. The proposed modeling approach uses Bayesian networks and rule-based methods that operate on state variables that have been carefully selected to represent the characterizing properties of groups and organizations. Specifically, state variables are used to represent the CGF's knowledge or beliefs about itself, other actors and the environment; Bayesian networks are used to model the behavior and action selection mechanism of the CGF; and rules are used to model actions and their effects on the CGF's state variables.

Furthermore, to reduce authoring complexity, this work also presents a sampling-based analysis method capable of analyzing the behavior and real-time performance of a model even prior to embedding the model in its intended simulation environment.

1 INTRODUCTION

Computer generated forces (CGFs) are autonomous, computer controlled, entities employed to model human actors in many simulation-based training, exercise, planning and decision support tools. Often CGFs are used to represent individual soldiers or teams of soldiers on a virtual battlefield [10]. Such CGF implementations are ideal for tactical simulators where the focus is on short-term scenarios involving only a handful of actors. However, battlefield simulations targeting the operational or strategic levels of warfare, where the focus is on long-term scenarios with thousands of actors, typically require other high-level and aggregate CGF models due to for instance limited computational resources or real-time requirements.

In this work we present a CGF modeling approach targeted towards modeling aggregate entities of human groups and organizations such as civilians, armed forces, etc. in military conflict zones. The proposed CGF modeling approach uses a combination of Bayesian networks and rule-based methods that operate on state variables that have been selected to represent the characteristic properties of aggregate entities representing groups and organizations. Specifically,

- state variables are used to represent the CGF's knowledge or beliefs about itself, other actors and the environment,
- Bayesian networks are used to model the behavior and action selection mechanism of the CGF,
- rules are used to model actions and their effects on the CGF's state variables.

CGF modeling is a complex task and it is often difficult, even for the modeler, to fully understand the dynamics of a model. In an attempt to reduce authoring complexity, we have in this work also developed a sampling-based analysis method that use Monte-Carlo simulation to generate data which can be statistically summarized and visualized to gain insight into the behavior and real-time performance of the model even prior to embedding the model in its intended simulation environment.

Although the modeling approach was originally intended for the development of CGFs in a real-time decision support tool for effects-based planning [16], the modeling and analysis methods as well as the implemented software modules can be adapted and reused in other, similar, simulation applications.

This document has been organized as follows. Related works are presented in Section 2. Our CGF modeling approach is presented in Section 3. In Section 4 we introduce and provide examples illustrating our sampling-based analysis method. Finally, conclusions are presented in Section 5.

2 RELATED WORKS

Alt and Lieberman introduce the Cultural Geography model (CG) which is a modular agent-based modeling (ABM) [4] approach towards representing human populations' behavioral response in irregular warfare operations [1, 2]. The purpose of the model is to gain *insight* into complex social systems by means of modeling and simulation ultimately resulting in improved course of action analysis (COA) tools and training simulators. Theories of narrative identity and planned behavior are used to model the cognition of each agent. Specifically, the CG model employs Bayesian networks (BNs) [9] to encode agent beliefs, values and interests. When an event or action occurs in the simulation, information about the event is automatically passed to agents within the vicinity of the event. Furthermore, homophily social network theory is used to propagate event information to agents outside the vicinity such that similar agents, using a metric of social distance, are more likely to communicate event information compared to those that are less similar. Given the event, each agent infers from the BNs its stance on a particular issue of interest (e.g., ”Is security adequate?” or ”Will the upcoming elections be legitimate?”). Using the model, a decision maker is able to execute many alternative COAs and select the one COA that resulted in the most satisfactory outcome with respect to selected issues of interest.

Sokolowski and Banks model the complex social behavior of civil uprising and insurgency using a system dynamics (SD) [11] approach in [17]. The purpose of the model is to study the British counter-insurgency in Ireland (1916, 1919-1921) and to better understand which factors that united the Irish insurgency and self-rule by means of *what-if* simulation. A narrative is used as an initial input to generate a causal loop diagram (CLD) describing the relevant variables and how these influence each other in the model. The CLD is then converted into a stocks-and-flow diagram (SFD) and a set of ordinary differential equations (ODEs) that govern the dynamics of the social system. The model includes a system of variables and equations describing the main actors: the general population, the insurgents (potential and recruited), and the British troops. Furthermore, the model uses variables modeling Irish satisfaction with British rule, coercive acts and insurgent incidents to highlight a few. The model was validated and calibrated using historical data. Yet another SD model of complex social behavior is presented by Grynkewich and Reifel in [12].

The MASON RebeLand model [7, 5] simulates the stability of a polity (i.e. state) when put under varying degrees of political stress. The purpose of the model is to study: 1) how a polity responds to various levels of societal stress and governmental performance; and 2) how insurgency, political instability emerges in a polity over time. The model consists of a combination of meso-scale models (state, city, general population, rebel groups) and micro-scale models (rebels, military units and police) where each model is implemented as one agent in the simulation. Issues (economical, environmental or security) are created and added to the simulation by RebeLand's socio-natural environment component. The RebeLand simulation was extended in [6] to include multiple states to study the dynamics of refugee flows, transnational conflict and crime, and natural hazards across borders.

REsCape [3] is an agent-based simulation framework that can be used as an exploratory tool to study the relationships between natural resources, ethnicity and civil war. The simulation is driven by a government that collects revenue from natural resources and applies different spending strategies which in turn affects the government support among the population. The population consists of peasant agents who may choose to follow the reigning government or to join a rebel group which employs a better spending strategy given the agent's self-interest. Conflicts occur if the rebel leaders and the government leaders attempt to control the same location on the grid-based map.

Irreducible Semi-Autonomous Adaptive Combat (ISAAC) and Enhanced ISAAC Neural Simulation Tool (EINSTein) are two simulation frameworks that specialize on studying emergent self-organizing behavior in warfare [14, 15]. Unlike the traditionally used *Lanchester Equation* (LE) method of studying warfare, the EINSTein framework employs an agent-based approach to facilitate spatial variation and specialized behaviors in the simulation. EINSTein is capable of capturing complex dynamics of warfare where the combatants are structured in teams that are capable of reacting and adapting to changing conditions in the environment.

3 MODELING APPROACH

3.1 State variables

State variables are used here to represent the CGF's knowledge and beliefs about itself, other actors and the environment. We have separated the state variables into sets representing an actor's internal state and its relationships to others. Note that in our CGF model all actors known to the CGF, including itself, are represented using separate sets of the abovementioned state variables. The purpose of the state variables is to provide a common knowledge representation that can be used when developing CGF behaviors and actions as described in Section 3.2 and Section 3.3. The state variables presented in this section were identified using subject matter experts and chosen to represent a wide range of characteristics among groups and organizations in military conflict zones.

The internal state variables, which originates from previous work presented in [16], are represented here by a vector, $\mathbf{I}$, that contains 16 discrete state variables. The name, label (A-P) and a brief (non-exhaustive) qualitative interpretation for each variable value is presented in Table 1. The variables in the internal state vector is limited to four integer values [0,...,3]. This design decision was made to keep the knowledge space of the CGF relatively small which ensures that behavior and action modeling remains pragmatic and not too time-consuming. Also, such limitation significantly reduces the complexity in terms of search space when embedding CGF models in real-time planning tools such as the one presented in [16].

Similarly to the internal state of an actor, its relationships to others are encoded in a relationship state vector, $\mathbf{R}$, as illustrated in Table 2. Each row in the table represents the relationship towards another actor. Note that, unlike $\mathbf{I}$, which is fixed, the number of variables in $\mathbf{R}$ varies with the number of other actors, N, known to the CGF. The relationship variables have four integer values [0,...,3] which are interpreted as *enemy*, *suspicious*, *neutral* and *friendly* respectively.

Given the state variables described above we can now introduce the notation used in the remainder of this paper. An actor, a_i, known to the CGF is represented by $\boldsymbol{\omega}_i = \{\mathbf{I}_{ij}, \mathbf{R}_{in}\}$ where $j = \{A, B, \ldots, P\}$ and $n = \{1, 2, \ldots, N\}$. That is, $\mathbf{I}_{ij}$ represents actor a_i's internal state variable j and $\mathbf{R}_{in}$ represents actor a_i's relationship to actor a_n. Given that the CGF knows about N actors (including itself) its complete knowledge space is $\boldsymbol{\Omega} = \{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$.

Let's also introduce the concept of roles that is used here to generalize action and behavior modeling: the *initiator* role is assigned to the actor that initiates an action; the *target* role is assigned to the actor who's state variables are directly affected by the *initiator*'s action; and the *bystander* role is assigned to all other

Table 1: Internal state variables.

State variable		Interpretation of state variable values 0	1	2	3
Weapon power	A	Less than 0.1 brigade	0.25 brigade	1 brigade	4 brigades
Living conditions	B	Suffering, scarce resources, being killed by others	Suffering, scarce resources	Not suffering, limited resources	Not suffering, abundant resources
Stance	C	Submissive	Defensive	Defiant	Violent
Sympathizers	D	No supporters	Supported by marginal others	Supported by the local majority	Supported by the wider majority
Economy	E	$<$ 1000 times GNP/capita	1000 $-$ 10000 times GNP/capita	10000 $-$ 100000 times GNP/capita	$>$ 100000 times GNP/capita
Stability	F	Quick reduction in group size	Slow reduction in group size	Slow increase in group size	Quick increase in group size
Geographical dominance	G	At risk in the area	Can move and talk freely	Can impose restrictions on others	Can dominate others
Infrastructure	H	Man-to-man, word-of-mouth	Terrain vehicles, leaflets	Trucks, cellphones	Complete, Internet
Propaganda channels	I	Limited reach outside primary group	Reaches local communities	Reaches communities of similar identity	Reaches all types of communities
Social network	J	No ties	Ties to uncommitted	Ties to committed	Ties to highly committed
Reputation	K	Despised	Light-weight	Recognized	Highly regarded
Dissatisfaction	L	No grievance	Would like to see responsible for grievance fail	Prepared to use violence in act of revenge	Prepared to sacrifice life in act of revenge
Group feeling	M	Power struggle	Friction	Harmony	Cohesive
Ideological conviction	N	None	Little	Medium	High
Goal orientation	O	None	Preserving	Advance	Vision
Moral stance	P	Indiscriminate use of violence	Low barrier/concern for out-groups	Restricted but pragmatic use of violence	Violence only as a last resort in self-defence

Table 2: Relationship state variables.

State variable		Interpretation of state variable values 0	1	2	3
Relationship$_1$	R$_1$	Enemy	Suspicious	Neutral	Friendly
⋮	⋮	⋮	⋮	⋮	⋮
Relationship$_n$	R$_N$	Enemy	Suspicious	Neutral	Friendly

actors, other than the *initiator* and *target*, that may be affected by the action. Henceforth, when referring to the state variables of the *initiator*, *target* and *bystander* actors the subscripts i, t and b are used respectively. For instance, $\mathbf{I}_{tA}$ refers to the internal state variable A of *target* actor a_t.

3.2 Modeling behaviors

Using the state variables we are now ready to introduce our behavior modeling method. The behavior of a CGF is in essence an action selection strategy implemented as a function that use as input the CGF's state variables, $\mathbf{\Omega}$, and generates as output the action, α, to execute as shown in Equation 1.

$$\alpha = f(\mathbf{\Omega}) \tag{1}$$

In this work a Bayesian network (BN) [9] approach has been adopted to model f using either subject matter expert knowledge in cases where too little or no data are available or using machine learning algorithms in cases where large data sets representing the historic behavior of an actor are available. We have primarily chosen to use BNs due to their:

- Capability to graphically represent CGF behavior using directed acyclic graphs (DAGs) which ultimately improves the general understanding of the model,
- Capability to perform inference, or select actions, even in the presence of missing or uncertain information,
- Modularity and re-usability.

The Bayes rule defined in Equation 2 represents the core of any Bayesian modeling approach [9]. Using the Bayes rule a probability value, the *posterior*, is calculated for each action available to the actor. Typically, the action with the maximum *posterior* is selected by the actor. This is however not always the case as will be discussed below.

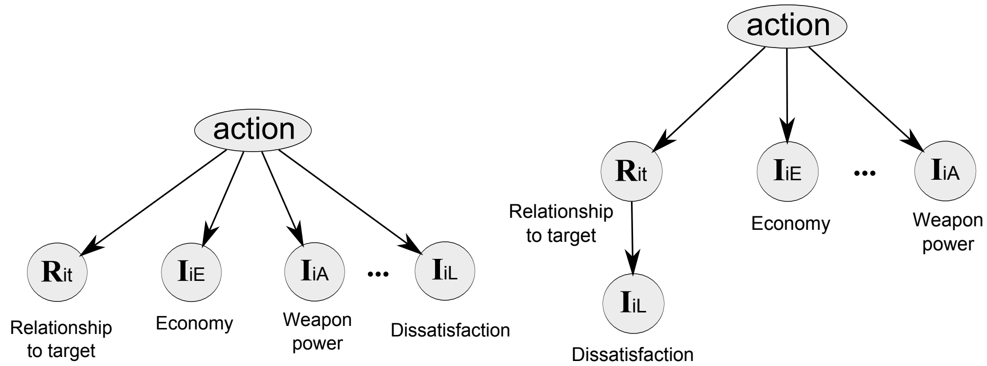
$$p(\alpha_n|\mathbf{\Omega}) = \frac{p(\alpha_n) \times p(\mathbf{\Omega}|\alpha_n)}{\sum_{m=1}^{M} p(\alpha_m) \times p(\mathbf{\Omega}|\alpha_m)} \tag{2}$$

From the Bayes rule it is clear that the *posterior*, $p(\alpha_n|\mathbf{\Omega})$, of action, α_n, is calculated using the *prior*, $p(\alpha_n)$, and likelihood, $p(\mathbf{\Omega}|\alpha_n)$, functions. The denominator, or the *evidence*, is a normalizing factor that spans all actions, M. That is, using the Bayesian approach it is ultimately the *prior* and *likelihood* functions that the modeler manipulates or that the learning algorithm estimates to represent desired actor behaviors. The problem with Bayes rule is that one rarely can find enough data to model the likelihood function due to, in our case, the high dimensional state variable vector $\mathbf{\Omega}$. This is where BNs comes to rescue by introducing conditional independence between variables, hence, simplifying the likelihood estimation process.

At its simplest a BN is identical to the naïve Bayes classifier in which all variables in $\mathbf{\Omega}$ are assumed to be conditionally independent of each other. Using this assumption, Bayes rule can be reduced to Equation 3, where K is the dimensionality of $\mathbf{\Omega}$. The DAG of an example naïve Bayes classifier BN is presented in Figure 1a.

$$p(\alpha_n|\mathbf{\Omega}) = \frac{p(\alpha_n) \times \prod_{k=1}^{K} p(\Omega_k|\alpha_n)}{\sum_{m=1}^{M} p(\alpha_m) \times \prod_{k=1}^{K} p(\Omega_k|\alpha_m)} \tag{3}$$

However, clearly not all variables in $\mathbf{\Omega}$ are conditionally independent of each other. As an example, an actor, a_i's, dissatisfaction, $\mathbf{I}_{iL}$, to another actor, a_t, is conditionally dependent on its perceived relationship, $\mathbf{R}_{it}$, to a_t. A modified network incorporating this conditional dependency is presented in Figure 1b. Links between any two variables in the DAG indicates that there exists a conditional dependency between them. Many inference algorithms that are capable of calculating the probabilities at arbitrary nodes in arbitrary structured BNs have been discussed in the literature [13]. In this study we have chosen to use the algorithm presented in [8]. It is important to know that the time required to infer probabilities varies depending on the structure of the BN as well as the amount of evidence (or knowledge) that are known prior to inference.



(a) Naïve Bayes classifier.

(b) BN classifier where I_{iL} is conditionally dependent on R_{it}. All other state variables are assumed to be independent of each other.

Figure 1: Examples of directed acyclic graphs representing Bayesian networks.

Using the probabilities inferred at the *action* node of the BN it is possible to select an action in several ways. Which action selection method to use is ultimately the modelers choice. This CGF model supports the following action selection methods:

- Maximum a posterior (MAP),
- Random draw.

The MAP approach, which is defined in Equation 4, simply selects the action with the maximum *posterior*. The random draw approach selects an action by randomly sampling the *posterior* values with respect to their proportions.

$$\alpha = \arg \max_{n \in \{1,\ldots,M\}} p(\alpha_n | \mathbf{\Omega}) \tag{4}$$

3.3 Modeling actions

Actions are the means by which a CGF may alter its state, $\mathbf{\Omega}$. An action is represented here by a set of rules each consisting of a condition, the *if*-part, and a list of effects, the *then*-part, such that if the condition is *true* then the list of effects will execute ultimately resulting in state variable changes. On the other hand, if the condition is *false* then none of the effects will execute.

In this work subject matter experts have developed hundreds of rules modeling the following actions: *attack*, *neutralize*, *negotiate*, *support*, *protect*, and *nothing*. In addition to the rules governing the effects of actions, subject matter experts have developed global rules modeling phenomena such as the Stockholm syndrome and radicalization. Global rules are also used to introduce constraints that filters out invalid or unwanted state variable values.

The conditions, or the *if*-parts, of the rules are described using Boolean expressions. The effects, or the *then*-parts, of the rules are described using a function notation where the *set*, *inc* and *dec* functions are used to set, increment and decrement specific state variable values. For instance, $set(\mathbf{I}_{tA}, 1)$ assigns the value 1 to the the *target* actor's, a_t, internal state variable, A. Similarly, $inc(\mathbf{I}_{tA}, 1)$ and $dec(\mathbf{I}_{tA}, 1)$ increases and reduces the same value by 1 respectively. Table 3 illustrates the notation using an example rule that partially models the effects of the *protect*-action.

Table 3: Example rule partially modeling the protect action.

If	Then	Description
$\mathbf{I}_{tA} > 0 \wedge \mathbf{I}_{tC} > 2 \wedge \mathbf{I}_{tD} > 1 \wedge \mathbf{I}_{tG} > 1 \wedge \mathbf{I}_{tI} > 0 \wedge \mathbf{I}_{tJ} > 1 \wedge \mathbf{I}_{tK} > 1 \wedge \mathbf{I}_{tL} > 1 \wedge \mathbf{I}_{tN} > 0 \wedge \mathbf{I}_{tO} > 1$	set($\mathbf{I}_{tB}$,2) set($\mathbf{I}_{tG}$,1)	Update living conditions and geographical dominance of target.

4 SAMPLING BASED ANALYSIS

To reduce the authoring complexity of our CGF modeling approach we have developed a sampling based analysis method that can be utilized to gain insight into the action selection and execution mechanisms of the CGF as well as its real-time performance. The analysis method is useful not only during CGF prototyping and development but potentially also for test and verification purposes.

The core of our analysis method consists of a Monte-Carlo based simulation environment where CGF implementations, or prototypes, can be embedded and executed. The simulation environment records the dynamics of the embedded CGF's state variables as well as all the actions that the CGF selected and executed. Following the Monte-Carlo method any input variables required by the CGF, to for instance invoke the action selection mechanism, are randomly sampled using pre-determined probability distributions.

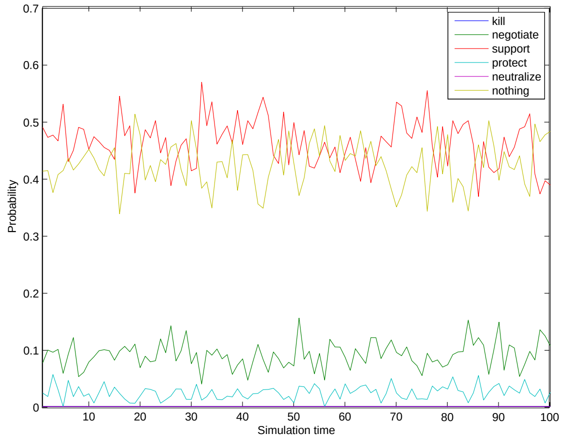
The pseudo-code of our simulation environment is presented in Algorithm 1. The algorithm basically consists of two loops where the inner loop randomly samples the *target* and *bystander* actors to be used as input to the CGF's action selection algorithm. Note that there is no need to sample the *initiator* role as it is always played by the CGF itself. When an action is executed in the inner-loop the state variables are updated, following the rules of the selected action, and the resulting state variables are copied and stored in a list for processing by the outer-loop. The outer loop creates a probability distribution using the list generated by the inner-loop. The probability distribution is randomly sampled to create a new state variable vector to be used in the next simulation cycle. This process is repeated for all simulation cycles.

Given the output of the MC-based simulation environment we can separate the analysis into three parts: 1) action selection analysis, 2) state variable analysis, and 3) execution time analysis. The action selection analysis is relevant to gain insight into the behavior or action selection mechanism of a CGF. Example plots illustrating the action selection behavior of a prototype CGF in the presence of six other fictitious actors (*BFOR*, *Compett*, *Nottovio*, *Popett*, *Popto* and *Poptre*) are provided in Figure 2. In Figure 2a the probabilities of selecting an action for a given target is presented over the entire simulation run. In Figure 2b and Figure 2c the probabilities of selecting actions are presented at a fixed point in time comparing three different knowledge levels (0%, 50% and 100%). The knowledge levels are used here to gain insight into how the action selection mechanism behaves in the presence of unknown information. When the knowledge level is at 0% all nodes (i.e. state variables) in the BN are unknown, at 50% values has been randomly added to 50% of the nodes and at 100% values has been added to all nodes. Figure 2b represents the CGF's action selection behavior for all actions applied to all *targets*. More detailed insights for a specific action, in this case the *attack* action, are provided in Figure 2c.

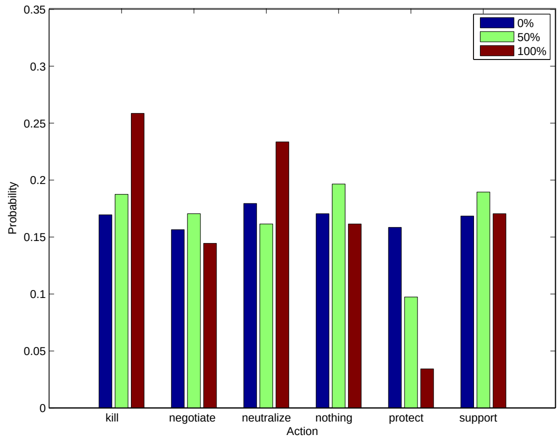
Algorithm 1 Monte-Carlo simulation algorithm for data generation.

Require: Initial state variables, $\mathbf{\Omega}$, simulation ticks, T, and Monte-Carlo iterations per simulation tick, K.

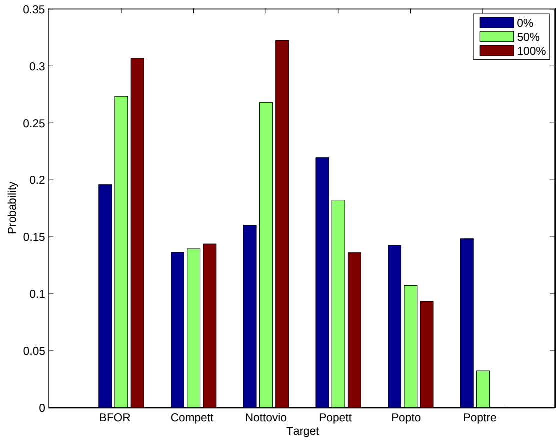
$SimulationData$ = empty list of states
Add $\mathbf{\Omega}$ to $SimulationData$
t = 0
while $t < T$ **do**
    $MonteCarloData$ = empty list of states
    k = 0
    while $k < K$ **do**
        $\mathbf{\Omega}_t$ = state at the end of the $SimulationData$ list
        a_t = randomly sample target actor
        a_b = randomly sample bystander actors
        α = select action by invoking the a_i's behavior module given $\mathbf{\Omega}_t$, a_t and a_b
        $\mathbf{\Omega}_k$ = calculate new state by executing α given $\mathbf{\Omega}_t$, a_t and a_b
        Add $\mathbf{\Omega}_k$ to $MonteCarloData$
        k = k + 1
    end while
    $\mathbf{\Omega}_{t+1}$ = randomly sample $MonteCarloData$
    Add $\mathbf{\Omega}_{t+1}$ to $SimulationData$
    t = t + 1
end while
return $SimulationData$



(a) The CGF's action selection probabilities at time interval 0-100 for a given target.



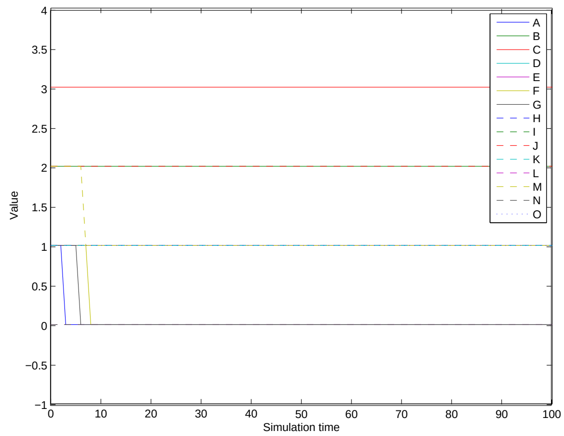
(b) The CGF's action selection probabilities at time=0.



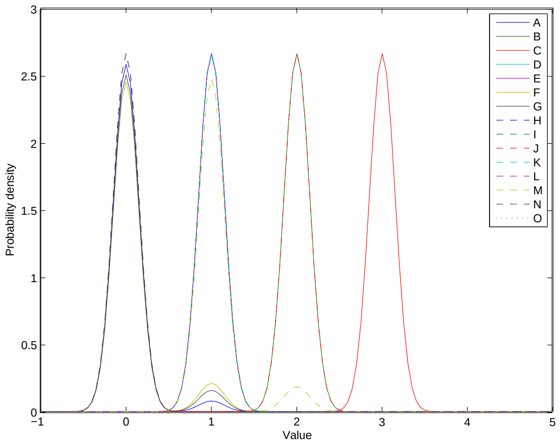
(c) The CGF's action selection probabilities for a given action at time=0.

Figure 2: CGF action selection probability plots. Probabilities can be plotted over time (Figure 2a) or at specific points in time (Figure 2b and Figure 2c). In Figure 2b and Figure 2c results are shown for CGF knowledge levels 0%, 50% and 100%.

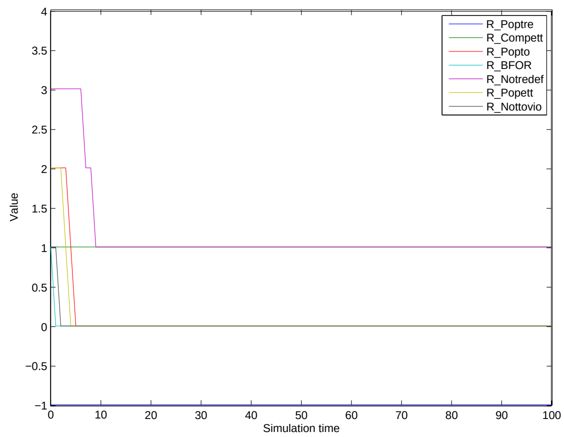
State variable analysis is relevant when developing actions but also to describe the conditions by which an action was selected by the CGF. In Figure 3 plots are provided to illustrate the CGF's knowledge of another actor's internal state and relationship variables over time. Line plots are used to gain insight into how state variable values changes throughout the simulation run. Probability density plots, which are created using non-parametric kernel density estimation [9], are provided here to complement the line plot to identify variation in state variable values.



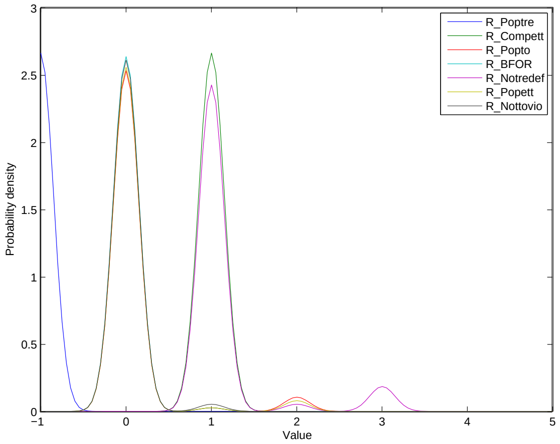
(a) Internal state.



(b) Internal state probability density.



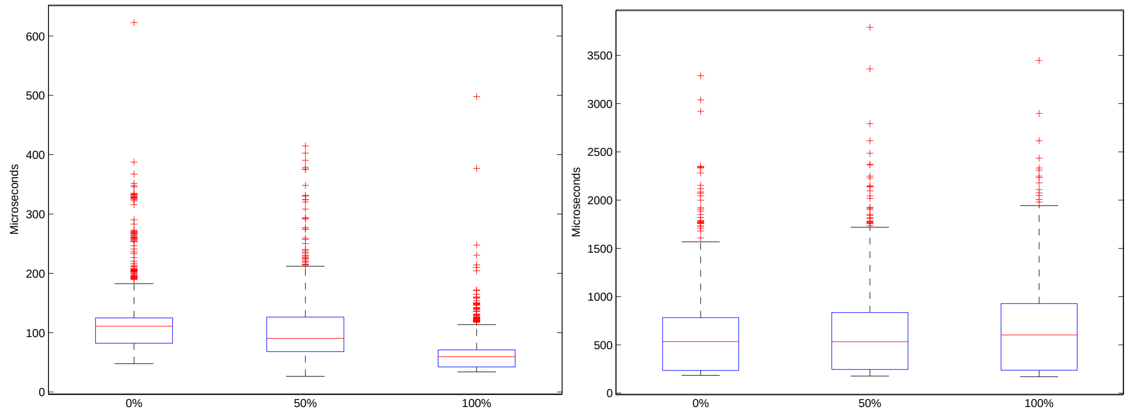
(c) Relationship state.



(d) Relationship state probability density.

Figure 3: Line and probability density plots illustrating the evolution of internal state and relationship variable values over time for a given actor.

Execution time analysis is useful to identify potential issues with respect to real-time requirements of the CGF. In Figure 4 box-plots are used to visualize the execution time of action selection as well as action execution. As mentioned above, the execution time depends on structure of the BN as well as the knowledge level of the CGF. To emphasize the impact of knowledge levels with respect to execution time, separate box-plots were generated for each of the previously defined knowledge levels.



(a) Time used for action selection.

(b) Time used for action execution.

Figure 4: Box-plots visualizing the execution time required by the CGF to select and execute actions. Results are shown for CGF knowledge levels 0%, 50% and 100%.

5 CONCLUSIONS

In this paper we have introduced a CGF modeling approach targeting the aggregate representation of groups and organizations in military conflict zones. Our approach is based on state variables, which have been identified by subject matter experts, to represent the CGF's knowledge and beliefs with respect to the characteristics of groups and organizations. Using the state variables a Bayesian network approach was proposed to model the behavior, or action selection mechanism, of the CGF and a rule-based approach was proposed to model the effects of actions.

Furthermore, given the complexity of CGF modeling, we have introduced a sampling based analysis method used in our work to gain insight into the behavior of CGF implementations even prior to embedding them in their intended simulation environment.

ACKNOWLEDGMENT

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.

REFERENCES

[1] Jonathan K. Alt, Leroy A. Jackson, David Hudak, and Stephen Lieberman. The cultural geography model: Evaluating the impact of tactical operational outcomes on a civilian population in an irregular warfare environment. *Journal of Defense Modeling and Simulation (Special Issue on Counter-Insurgency)*, 2010.

[2] Jonathan K. Alt and Stephen Lieberman. The cultural geography model: An agent based modeling framework for analysis of the impact of culture in irregular warfare. *International Conference on Computational Cultural Dynamics*, 2011.

[3] Ravi Bhavnani, Dan Miodownik, and Jonas Nart. Rescape: an agent-based framework for modeling resources, ethnicity, and conflict. *Journal of Artificial Societies and Social Simulation*, 11(2):7, 2008.

[4] Eric Bonabeau. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):7280–7287, 2002.

[5] Claudio Cioffi-Revilla. Simplicity and reality in computational modeling of politics. *Computational and Mathematical Organization Theory*, 15:26–46, 2009. 10.1007/s10588-008-9042-2.

[6] Claudio Cioffi-Revilla and Mark Rouleau. MASON AfriLand: a regional multi-country agent-based model with cultural and environmental dynamics. *Proceedings of the Human Behavior-Computational Modeling and Interoperability Conference (HB-CMI-09)*, 2009.

[7] Claudio Cioffi-Revilla and Mark Rouleau. MASON RebeLand: an Agent-Based model of politics, environment, and insurgency. *International Studies Review*, 12(1):31–52, 2010.

[8] Fabio Gagliardi Cozman. Generalizing variable elimination in bayesian networks. In *In Workshop on Probabilistic Reasoning in Artificial Intelligence*, pages 27–32, 2000.

[9] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[10] Rick Evertsz, Matteo Pedrotti, Paolo Busetta, Hasan Acar, and Frank E. Ritter. Populating VBS2 with realistic virtual actors. *Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation BRIMS*, pages 1–8, 2009.

[11] Jay Forrester. Industrial Dynamics: A major breakthrough for decision makers. *Harvards business review*, 36(4):37–66, 1958.

[12] Alex Grynkewich and Chris Reifel. Modeling Jihad: A System Dynamics Model of the Salafist Group for Preaching and Combat Financial Subsytem. *Strategic Insights*, 5(8), 2006.

[13] Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15:225–263, 1996.

[14] Andrew Ilachinski. Irreducible semi-autonomous adaptive combat (ISAAC): An artificial life approach to land combat. *Military Operations Research*, 5:29–46, 1997.

[15] Andrew Ilachinski. *Artificial War: Multiagent-Based Simulation of Combat*. World Scientific Press, 2004.

[16] Farshad Moradi and Johan Schubert. Modelling a simulation-based decision support system for effects-based planning. *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)*, (11):1–14, 2009.

[17] John A. Sokolowski and Catherine M. Banks. Modeling complex social behavior: A system dynamics approach. *Proceedings of the 19th Conference on Behavior Representation in Modeling and Simulation*, March 2010.