Biuletyn WAT
Vol. LVII, Nr 4, 2008

Modelling and simulation of C^2 processes based on cases in the operational simulation system for CAXes

RYSZARD ANTKIEWICZ, WOJCIECH KULAS, ANDRZEJ NAJGEBAUER, DARIUSZ PIERZCHAŁA, JAROSŁAW RULKA, ZBIGNIEW TARAPATA, ROMAN WANTOCH-REKOWSKI

Military University of Technology, Cybernetics Faculty, Institute of Information Systems, 2 Kaliskiego Str., 00-908 Warsaw

Abstract. The Simulation Based Operational Training Support System (SBOTSS), which was constructed in order to provide cost-effective approach of Computer Assisted Exercises, is an integrated, interactive, many-sided land analysis and training support model (with logistics, engineering, electronic warfare and intelligence functions).
The idea and model of command and control process applied for the decision automata on the tactical level are presented. The automata execute the two main processes: decision planning process and direct combat control. The decision planning process relating to the automata contains three stages: identification of a decision situation, generation of decision variants (action plans), variants evaluation and nomination of the best variant of these, which satisfy the proposed criteria. The particular approach to identification of decision situation and variants of action are presented. The procedure of variants generation, based on some kind of pre-simulation process, contains the evaluation module, which allows us the best choice of action plan according to specified criteria. The direct combat control process contains such phases like command, reporting and reaction to fault situations. Some results of the simulation process including the decisions made by automata are considered.
Keywords: combat modelling and simulation, decision automata, system for CAX
Universal Decimal Classification: 355.077

Introduction

The idea of military unit structure used in SBOTSS is presented in Fig. 1.

Physically simulated objects in this structure are command posts, automatic commanders and basic units (in the SBOTSS basic units means company, battery

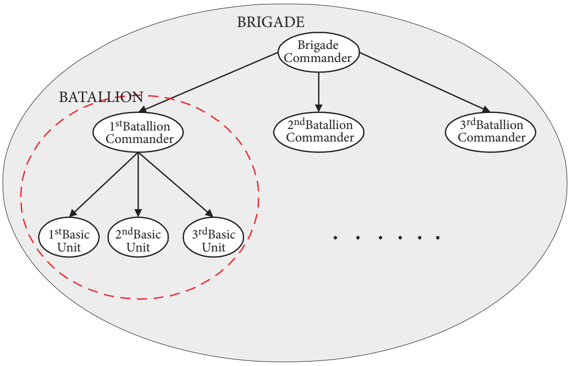

Fig. 1. The structure of simulation model of brigade

or logistic platoon). The source of effectiveness is in the limitation of staff personnel, required in the CAXs and replacement of the staff by "automatic commanders".

The decision situation is classified according to the following factors: own task, expected actions of opposite forces, environmental conditions — terrain, weather, the day and year season, current state of own and opposite forces in a sense of personnel, weapon systems and military materiel. For each class of decision situation there is generated the set of action plan templates for subordinate and support forces. For example, the proposed action plan contains: forces redeployment, regions of attack or defence, or manoeuvre routes, intensity of fire for different weapon systems, terms of supply of military materiel combat forces by logistics units. In order to generate and evaluate possible variants we use the pre-simulation process based on some procedures — forces attrition procedure, slowing down rate of attack procedure, utilization of munitions and petrol procedure.

In the evaluation process we consider the following criteria: time and degree of task realization, own losses, utilization of munitions and petrol. The idea of decision generation using 3-stage algorithm was presented in [2]. The presented paper is the continuation of the approach.

Model of decision situation

The model of a decision situation concerns the first two steps (elliptical line) in Fig. 2. We define decision situations space as follows:

$$DSS = \{SD: \ SD = (SD_r)_{r=1,..,8}\}.$$

The vector *SD* represents the decision situation which is described by the following eight elements:

— SD_1 — the commanding level of opposite forces,

— SD_2 — the type of task of opposite forces (e.g. attack, defence),
— SD_3 — the commanding level of opposite forces,
— SD_4 — the type of a task of own forces (e.g. attack, defence),
— SD_5 — the net of squares as a model of activities (interest) area

$$SD_5 = \left[SD_{ij}^5 \right]_{\substack{i=1,..,SD_7 \\ j=1,..,SD_8}},$$

— SD_7 — the width of activities (interest) area (number of squares),
— SD_8 — the depth of activities (interest) area (number of squares),

$$SD_{ij}^5 = (SD_{ij}^{5,k})_{k=1,..,6},$$

where for the terrain square with the indices (*i*, *j*) each of elements denotes:

— $SD_{ij}^{5,1}$ — the degree of the terrain passability,
— $SD_{ij}^{5,2}$ — the degree of topographical terrain configuration,
— $SD_{ij}^{5,3}$ — the degree of terrain growth,
— $SD_{ij}^{5,4}$ — the armoured power (potential) of opposite units deployed in the square,

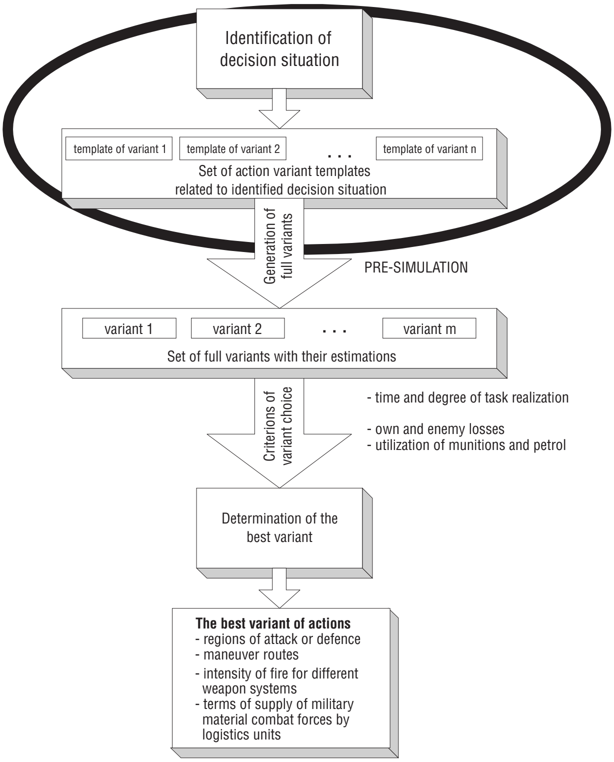

Fig. 2. Algorithm for selecting the best variant of action [1]

— $SD_{ij}^{5,5}$ — the infantry power (potential) of opposite units deployed in the square,
— $SD_{ij}^{5,6}$ — the artillery power (potential) of opposite units deployed in the square,
— SD_6 — the description of own forces:

$$SD_6 = \left(SD_i^6\right)_{i=1,..,4},$$

— SD_1^6 — the summary armoured power (potential) of own units,
— SD_2^6 — the summary infantry power (potential) of own units,
— SD_3^6 — the summary artillery power (potential) of own units,
— SD_4^6 — the summary air fire support power (potential).

Problem of pattern identification for decision situations

We have the set of decision situation patterns: $PDSS = \{PS : PS \in DSS\}$. For the current decision situation $CS \in DSS$ we have to find the most similar situation from the set of patterns. Using the similarity measure function F_{CS} (see further part of this section) we can evaluate distances between two different decision situations, especially the current and the pattern. There are several methods of finding the most matched pattern situation to current one which can be used. We propose two main approaches dealt with the following measures: distance vectors measure and weighted graphs similarity measure.

We determine the subset of the decision situation patterns $PDSS_{CS}$ which are generally similar to the current situation considering such elements like: task type, command level of own and opposite units and own units potential

$$PDSS_{CS} = \{PS = (PS_i)_{i=1,..,6} \in PDSS : PS_i = CS_i,$$
$$i = 1,..,4, dist_{potwl}(CS, PS) \leq \Delta Pot\}$$

where

$$dist_{potwl}(CS, PS) = \max\{\left|CS_k^6 - PS_k^6\right|, k = 1,..4\}$$

ΔPot — the maximal difference of own forces potential.

Then, we formulate and solve the multicriteria optimization problem which allows us to determine the most matched pattern situation from the point of view of terrain and military power characteristics:

$$Z = \left(PDSS_{CS}, F_{CS}, R_D\right)$$

$$F_{CS}: PDSS_{CS} \to R^2$$

$$F_{CS}(PS) = (dist_{ter}(CS,PS), dist_{pot}(CS,PS))$$

$$dist_{ter}(CS,PS) = \sum_{k=1}^{3} \lambda_k \cdot \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \left(CS_{ij}^{5,k} - PS_{ij}^{5,k} \right)^p \right)^{\frac{1}{p}}$$

$$\sum_{k=1}^{3} \lambda_k = 1,\ \lambda_k > 0,\ k = 1,..,3$$

$$dist_{pot}(CS,PS) = \sum_{k=4}^{6} \mu_k \cdot \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \left(CS_{ij}^{5,k} - PS_{ij}^{5,k} \right)^p \right)^{\frac{1}{p}}$$

$$\sum_{k=4}^{6} \mu_k = 1,\ \mu_k > 0,\ k = 4,..,6$$

$$I = \min\{CS_7, PS_7\},\ J = \min\{CS_8, PS_8\}$$

$$R_D = \left\{ \begin{array}{l} (Y,Z) \in PDSS_{CS} \times PDSS_{CS}: \\ dist_{ter}(CS,Y) \le dist_{ter}(CS,Z) \wedge \\ dist_{pot}(CS,Y) \le dist_{pot}(CS,Z) \end{array} \right\}$$

For the hypothetical decision situations (*CS* — current, *PS* — pattern) presented in Fig. 3, the most matched pattern decision situation to the current situation *CS* using the above presented method is PS_2.

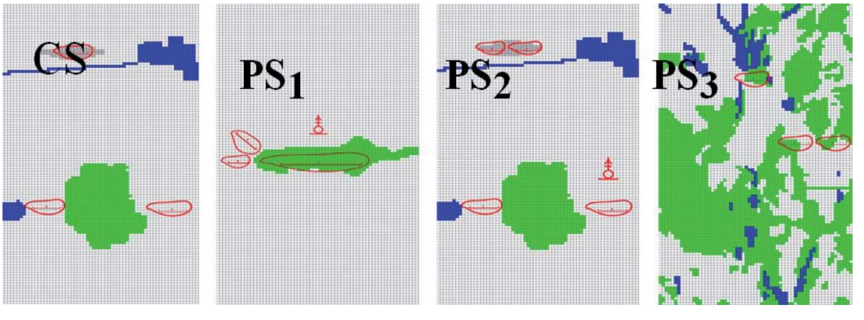

Fig. 3. Hypothetical current situation *CS* and pattern situations (PS_1, PS_2, PS_3)

In the literature, there are several methods for determining graphs similarity (based on: graphs isomorphism [3], graphs homeomorphism [3], adjacency matrices similarity [4]).

In our proposition, the graphs similarity approach for identification of the decision situation consists of three stages [5]:

(1) Building the weighted graphs $G(CS)$ and $G(PS)$ representing the decision situations: current (CS) and pattern (PS);
(2) Calculating the similarity measure $c(AS,PS)$ between the graphs $G(CS)$ and $G(PS)$;
(3) Selecting the most similar pattern situation to current situation.

Stage 1

The first stage is to build the weighted graphs G_T and G_D which describe decision situation (current and pattern). The graph G (G_T or G_D) is defined as follows:

$$G = \langle V, A \rangle,$$

where: V — the set of graph's nodes,
A — the set of graph's arcs, $A \subset V \times V$.

Each node of G describes terrain cells with non-zero values of characteristics defined as components of SD_{ij}^{5} defined in previous section. On each node of G we describe some functions which identify some part of decision situation regarding the considered terrain cell, e.g.: topographical conditions for the graph GT (degree of growth of: forests, waters, buildings etc., similar to: $SD_{ij}^{5,1}$, $SD_{ij}^{5,2}$, $SD_{ij}^{5,3}$, units deploying for the graph GD (location, military power, similar to: $SD_{ij}^{5,4}$, $SD_{ij}^{5,5}$, $SD_{ij}^{5,6}$. Two nodes $v1$, $v2 \in V$ are linked using arc $a \in A$ when the cells represented by $v1$ and $v2$ are adjacent (they are adjacent taking into account action direction, see Fig. 4). For example, in Fig. 4 we have terrain divided into 15 cells (3 rows and 5 columns). In some cells we have the units denoted by circles on the left-hand side. Structural representation of units deploying is defined by the graph GD (right-hand side).

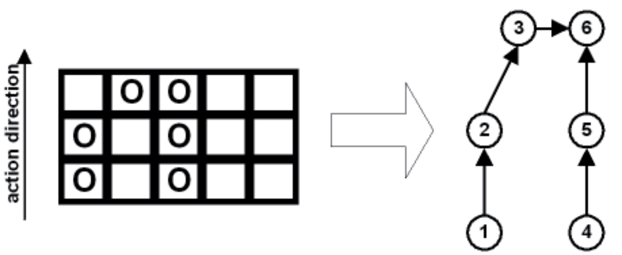

Fig. 4. Units deploying and their structural (graph) representation

Let's note that similar representation like in Fig. 4 we also have for topographical conditions (one graph for one of the topographical information layer: waters, forests, buildings or one graph GT for all of these information).

The pattern generation from the terrain point of view is based on the specific classification. The decision situation is classified according to the following factors: own task, expected actions of opposite forces, environmental conditions — terrain, weather, the day and year season, current state of own and opposite forces in a sense of personnel, weapon systems and military materials. The terrain classification method is based on some model of the terrain, which is used in the SBOTSS. This model is closely integrated with a geographic information system (digital map) and a simulation system and it is defined as regular grid of terrain squares. Regular grid of squares divides a terrain space into the squares with the same size and each square is treated as having homogeneity from the point of view of terrain characteristics (degree of velocity weakness, ability to camouflage, degree of visibility, etc.). This model is used to plan off-roads (cross-country) movement e.g. during attack planning. In the simulation system the second terrain model (as road-railroad network model) is also defined but this is not used in terrain classification method.

Terrain classification depends on the following characteristics:

— Terrain Topography (TT) = (surface, vegetation, soil);
— Weather (W) = (temperature, wind, precipitation, transparency):
  - o Temperature (WT) — high, medium, low,
  - o Wind (WW) — strong, medium, weak,
  - o Precipitation (WP) — strong, medium, lack,
  - o Transparency (WTR) — good, weak, bad;
— The season of the day (SD) — night, day (morning, afternoon, evening);
— The season of the year (SY) — spring, summer, autumn, winter.

The idea of the terrain classification method is to estimate a terrain region in which own and opposite units will operate to obtain one of the four kinds of the terrain: go, slow go, no go, no move.

The first kind of the terrain (go) is excellent for movement (e.g. plain terrain), the second one (slow go) is good for movement (e.g. soft-hilly terrain), the third kind of the terrain (no go) is poor for movement (e.g. hard-hilly terrain or mountainous terrain) and the last kind of the terrain (no move) describes impassable terrain (e.g. lakes, seas, high mountains).

The region (action strip) in which own and opposite units will operate is divided into the rectangular or trapezoidal subregions (each of these for subordinate unit). Inside each of the subregions and between adjacent subregions we determine the shortest paths from the start of the region to the end of it (the start of the region is taken from the side of own units and the end of the region is taking from the side of opposite units).

Stage 2

Having the weighted graphs $G_D(CS)$ and $G_D(PS)$ representing current and pattern decision situations (units deploying layer) we can modify graphs similarity approach [3, 4] to find the most similar decision situation pattern to current situation (for pair of the graphs $G_T(CS)$ and $G_T(PS)$ by analogy). The similarity is calculated as structural and non-structural (quantitative) similarity. This is the essence of modification of approaches presented in [4] and it is more precisely described in [5].

To calculate structural similarity between current and pattern situations represented by $G_D(CS)$ and $G_D(PS)$ we propose to use approach defined by Blondel, van Dooren et al. in [4].

In paper [2], the sequence of the matrices Z_{k+1} is calculated. Let C and P define the transition matrix of nodes for the graph $G_D(CS)$ and $G_D(PS)$. We calculate the following sequence of matrices:

$$Z_{k+1} = \frac{PZ_k C^T + P^T Z_k C}{\left\| PZ_k C^T + P^T Z_k C \right\|_F}, \quad k > 0,$$

where: $Z_0 = \mathbf{1}$ (matrix with all elements equal to 1);
x^T — the matrix x transposition;
$\|x\|_F$ — Frobenius (Euclidian) norm for the matrix x and
$\|x\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}^2}$, n — the number of matrix rows (number of nodes of $G_D(CS)$), m — the number of matrix columns (number of nodes of $G_D(PS)$).

We obtain the similarity matrix S^1 of graphs $G_D(CS)$ and $G_D(PS)$ nodes as follows:

$$S^1 = \lim_{k \to \infty} Z_{2k}$$

The similarity described by elements of the matrix S^1 is called "structural similarity". The element s_{ij}^1 of the matrix S^1 defines a normalized measure of similarity between the i-th node of $GD(CS)$ and the j-th node of $GD(PS)$. The greater value of s_{ij}^1, the greater similarity between the i-th node of $GD(CS)$ and the j-th node of $GD(PS)$. The essence of graph's nodes similarity is: two graph nodes are similar if their neighbourhoods are similar.

To calculate non-structural similarity between current and pattern situations represented by $GD(CS)$ and $GD(PS)$ we can calculate the distance matrices S^2 and S^3 between nodes of $GD(CS)$ and $GD(PS)$ from the point of view of the units locations (S^2) and the units military power (S^3) (functions f1 and f2 described on the graph's nodes) [2].

Finally, we build the matrix S which element s_{ij} is calculated as follows:

$$s_{ij} = \sum_{k=1}^{3} s_{ij}^k \cdot \lambda_k, \quad \sum_{k=1}^{3} \lambda_k = 1, \qquad \underset{k=1,\dots,3}{\forall} \lambda_k \in [0,1]$$

and combines structural and non-structural similarity.

Example of using graphs similarity approach to find the most matched pattern decision situation to current situation is presented in Fig. 5.

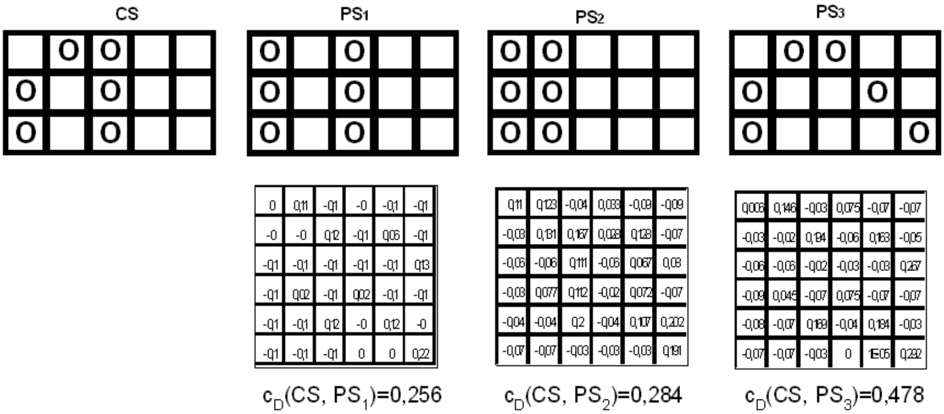

Fig. 5. Similarity matrices *S* between current decision situation *CS* and pattern situations PS_1, PS_2, PS_3

We set s_{ij} of *S* as follows: $s_{ij} = s_{ij}^1 \cdot 0.5 - s_{ij}^2 \cdot 0.5$.

Having the matrix *S*, we solve assignment problem (using e.g. Hungarian algorithm) to find the best allocation matrix $X=[x_{ij}]_{mxn}$ of nodes from a graph describing *CS* and *PS*:

$$c_D(CS,PS) = \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \cdot x_{ij} \rightarrow \max$$

with constraints:

$$\sum_{i=1}^{n} x_{ij} \leq 1, \quad j = \overline{1,\ldots,m}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \quad i = \overline{1,\ldots,n} \quad .$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} = \min\{n,m\}$$

$$x_{ij} \in \{0,1\}$$

The value $c_D(CS,PS)$ is the similarity measure of the graphs *GD*(*CS*) and *GD*(*PS*).

Stage 3

Having the set $PDSS_{CS}$ of pattern situations we calculate a value of the measure $c_D(CS, PS)$ for each $PS \in PDSS_{CS}$ and we select such PS* for which the following condition is satisfied:

$$c_D(CS, PS^*) = \max_{PS \in PDSS_{CS}} c_D(CS, PS).$$

For example, for situations presented in Fig. 5 we obtain that the most matched *PS* for *CS* is PS_3.

Having $c_D(CS, PS)$ and $c_T(CS, PS)$ we can find inside the set $PDSS_{CS}$ nondominated *PS* (taking into account two criteria c_D and c_T) or we can build metacriteria function using $c_D(CS, PS)$ and $c_T(CS, PS)$ to select the most matched decision situation pattern to current situation.

Generation of decision variant

Having pattern decision situation most similar to current situation, we could obtain a set of action plan templates from tactical knowledge base. Action plan template contains such elements as: type of formation, tasks of units in each echelon of formation, type of manoeuvre. In order to generate full operation plan, we should determine deployment of our forces, manoeuvre routes, plan of fire, tasks for support units and for air support, plan of supply of military materiel by logistic units.

The next steps, after generation of a set of operation plans, are evaluation of all variants of operation plan and the best choice of them. For variants evaluation we use the pre-simulation process based on some procedures: forces attrition procedure, slowing down rate of attack procedure, utilization of munitions and petrol procedure.

Forces attrition procedure is based on the following relations [2]:

$$Pog_B(id',t+\Delta t) = Pog_B(id',t) - \sum_{id \in JW_B^{-1}(id',t)} f\_\mathrm{int}(id,t) \cdot \Lambda_{ref}(id,t_0,id',dist(id,t,id')) \cdot \frac{Pog_A(id,t)/dist(id',t,id)}{\sum_{id'' \in JW_A(id.t)} (Pog_B(id'',t)/dist(id,t,id''))} \cdot \frac{Pog_A(id,t)}{Pog_A(id,t_0)} \cdot \Delta t$$

for $id' \in B$

$$Pog_A(id,t+\Delta t) = Pog_A(id,t) - \sum_{id' \in JW_A^{-1}(id',t)} f\_\mathrm{int}(id',t) \cdot \Lambda_{ref}(id',t_0,id,dist(id',t,id)) \cdot \frac{Pog_A(id,t)/dist(id',t,id)}{\sum_{id'' \in JW_B(id'.t)} (Pog_B(id'',t)/dist(id',t,id''))} \cdot \frac{Pog_B(id,t)}{Pog_B(id,t_0)} \cdot \Delta t$$

for $id \in A$,

where: A, B — the sides of combat,

$Pog_A(id,t), Pog_B(id,t)$ — the combat potential of two sides units,

$\Lambda_{ref}(id',t_0,id,dist(id',t,id)$ — the intensity of id' unit fire against the unit id, under distance condition $dist(id',t,id)$ and fully supplied units,

$f\_\mathrm{int}(id',t)$ — the part of full potential fire of unit id' used at the time t.

The slowing down rate of attack procedure uses the following functions:

$$v_{akt}(id,t) = \min\{v_{dec},\ v_{\max}^{op}(id,t)\}$$

where: $v_{\max}^{op}(id,t)$ — the real maximal velocity of the unit id;

$$v_{\max}^{op}(id,t) = v_{\max}(id,t) \cdot StOsl\Pr edk(Cond\_env(id,t), StSp_A(id,t), in\_kill\_ratio(id, JW_A^{-1}(id,t),t))$$

$\nu_{\max}(id,t)$ — the maximum velocity of the unit id depends on technical possibilities of armaments,

$StOsl\Pr edk$ — slowing down velocity function depends on:

a) terrain conditions — $Cond\_env(id,t)$,

b) unit percent dismounted — $StSp_A(id,t)$,

c) kill ratio index $in\_kill\_ratio(id, JW_A^{-1}(id,t),t)$ — depends on attrition rates of combat potential.

The utilization of munitions and petrol procedure is based on the following formulas:

— utilization of munitions

$$StSBiM_A(id,t+\Delta t,k) = \max\{0,\ StSBiM_A(id,t,k) - \frac{Pog_A(id,t)}{Pog_A(id,t_0)} \cdot f\_\mathrm{int}(id,t) \cdot \sum_{i \in SO:am(i,k)=1} \sum_{j \in SO(i)} sf(id,t_0,i,j) \cdot \overline{\lambda_{ij}}(t) \cdot \Delta t\}$$

— utilization of petrol

$$StMPiS(id,t+\Delta t,k) = \max\{0,\ StMPiS(id,t,k) - Z_{MPS}(id,t,\Delta t,k)\}$$

$$Z_{MPS}(id,t,\Delta t,k)=v_{akt}(id,t)\cdot\Delta t\cdot\frac{Pog_A(id,t)}{Pog_A(id,t_0)}\cdot$$
$$\cdot\sum_{\substack{i\in UISW,\, j\in UISW(i):\\ mps(i,k)=1,\, swp_A(id,t_0,i,j)\neq 0}} swp_A(id,t_0,i,j)\cdot bb\cdot(1+ke(id,t))$$

$$bb=\frac{(1+kmt(i,j,RDZJW(id,t),RDZJW(id',t),StPOb(id',t)))}{S(i,j)}$$

Manoeuvre routes and units velocity are determined using procedures, which contain two main parts:

- the determination of the shortest path for subordinate units under attack condition with maximum possible velocity,
- the modification of velocity values due to coordination of subordinate units during their actions on battlefield.

During pre-simulation process, we obtain values of such combat characteristics as: time and degree of task realization, own and enemy losses, utilization of munitions and petrol. Now we can formulate problem of finding the best operational plan as a multicriterion optimization problem with lexicographical relation. The next phase of automata activity there is direct combat control, which is connected with realization of decision made in previous phase. On the basis of observed actions of subordinate units, the automata react to the possible deviation of real trajectories in comparison to determined in a planning phase.

The implementation

The automata were implemented in environment of a distributed interactive simulation system in ADA language and it was tested with some scenarios of land combat exercises on a brigade level. The environment proposed is constructed as a distributed interactive simulator with respect to HLA (High Level Architecture). HLA was developed by the DMSO of the US DoD to meet the needs of reusability and interoperability in virtual, constructive and live simulations. Due to HLA features there is easy way to include new models, unit structures and tactical rules. The synchronization and communication mechanisms rely on conservative algorithms and implement assumptions of a constructive discrete-event simulation. Special extensions of ADA language were constructed to manage a set of simulation events, activities and simulation time. Time management services concern the chronological order of events (local and delivered to federates via messages), and the mechanisms for advancing simulation time.

The implementation of automata in ADA language (SBOTSS) contains many procedures:

```
type Variant_Of_Attack.Object is record
   -— in each of 2 strips of attack
   Potential_Ratio : Array_Of_Float;;
   Execution_Time : Array_Of_Float;
   Relative_Own_Potential : Array_Of_Float;
   Own_Potential_Losses : Array_Of_Float;
   Relative_Enemy_Potential : Array_Of_Float;
   Enemy_Potential_Losses : Array_Of_Float;
   Own_Initial_Potential : Array_Of_Float;
   Sum_Ammunition_Consumption : Float;
   Sum_Fuel_Consumption : Float;
   Sum_Losses_ratio : Float;
  Acceptable : Boolean;
  Group : Access_Array_Of_Array_Of_Integer;
end record;

procedure Determine_Tasks_For_Subunits(This :
         Compound_Unit.Handle) is
   Tasks : Task.Access_Array_Of_Handle renames
         This.Dec.Tasks;
   Variant : Variant_Of_Attack.Object;
   Sa : R_Situation_Assessment renames
         This.Situation_Assessment;
begin
   Variant := Determine_Best_Variant_Of_Attack(This =>
         This);
   Tasks := new Task.Array_Of_Handle(Sa.Own_Units'Range);
   for K in Tasks'Range loop
      Assigne_Task_To_Subunit(Sa.Own_Units(K), Variant,
         Tasks(K));
   end loop;
end Determine_Tasks_For_Subunits;

function Determine_Best_Variant_Of_Attack(This :
      Compound_Unit.Handle) return
      Variant_Of_Attack.Object is
   Variants : Variant_Of_Attack.Access_Array_Of_Object;
   Best_Variant_No : Integer;
begin
   Assess_Situation (This => This);
Unit_Variants (Variants => Variants);
   Evaluate_Variants (This => This,
      Variants => Variants.all);
   Best_Variant_No := Choose_Variant(Variants => Variants);
   return Variants(Best_Variant_No);
end Determine_Best_Variant_Of_Attack;

procedure Assess_Situation (This : Compound_Unit.Handle) is
   Sa : R_Situation_Assessment renames
      This.Situation_Assessment;
```

```
begin
  Divide_Strip_Of_Attack_By_2 (This);
  Utils.Determine_Own_Units (This);
  Utils.Evaluate_Ammunition_And_Fuel_Level (This);
  for K in 1 .. 2 loop
    Sa.Teren := Terrain_Classification(Strip_Of_Attack,
      V_Max => 60.0, Sub_Strip_No => K);
    Utils.Determine_Enemy_Units (This => This,
      Sub_Strip_No => K);
    Sa.Enemy_Defence_Preparation_Level(K) :=
      Evaluate_Enemy_Defence_Preparation_Level(K);
  end loop;
  Evaluate_Sum_Own_Units_Potential(Sa);
  Evaluate_Enemy_Initial_Potential (Sa);
end Assess_Situation;

procedure Evaluate_Variants(This : Compound_Unit.Handle;
Variants : in out Variant_Of_Attack.Array_Of_Object) is
  Number_Of_Accept_Variants : Integer := 0;
begin
  for K in 1 .. 2 loop
    for I in Variants'Range loop
      Evaluate_Own_Initial_Potential(Variant =>
          Variants (I), This => This, Sub_Strip_No => K);
      Evaluate_Potential_Ratio(This => This,
          Variant => Variants (I), Sub_Strip_No => K);
      Evaluate_Execution_Time(This => This,
          Variant => Variants (I), Sub_Strip_No => K);
      Evaluate_Losses(This => This,
          Variant => Variants (I), Sub_Strip_No => K);
      Evaluate_Amo_Fuel_Consumption(This => This,
          Variant => Variants (I), Sub_Strip_No => K);
  end loop;
end loop;
Evaluate_Variants_Execution_Time(This => This,
      Variants => Variants, N_Of_Accept_Variants =>
      N_Of_Accept_Variants);
if N_Of_Accept_Variants > 1 then
  Evaluate_Variants_Losses (This => This, Variants =>
      Variants, N_Of_Accept_Variants =>
      N_Of_Accept_Variants);
  Evaluate_Variants_Amo_Fuel_Consumption (This => This,
      Variants => Variants);
  end if;
end Evaluate_Variants;

-— procedure of unusual situation service
procedure Command_And_Control(This : Compound_Unit.Handle;
Report : Report.Handle) is
begin
  case Report.Event is
    when Ev_Lack_Of_Fuel =>
      Service_Lack_Of_Fuel(This => This);
    when Ev_Lack_Of_Ammo =>
```

```
        Service_Lack_Of_Ammo(This => This);
     when Ev_No_Move_Possibility =>
        Service_No_Move_Possibility(This => This);
     when Ev_Detection_Of_Contamination =>
        Service_Detection_Of_Contamination(This => This);
     when Ev_Detection_Of_Min_Field =>
        Service_Detection_Of_Min_Field(This => This);
     when Ev_End_Of_Task =>
        Service_End_Of_Task(This => This);
     when Ev_Rout_Of_Subunit =>
        Service_Rout_Of_Subunit(This => This);
     when Ev_Detection_Of_Enemy_Unit =>
        Service_Detection_Of_Enemy_Unit(This => This);
     when Ev_Enemy_Fire =>
        Service_Enemy_Fire(This => This);
     when others =>
        null;
  end case;
  Report_To_Superior(This => This);
end Command_And_Control;
```

Conclusions

The methods were implemented and tested for a different scenario. The automata realise their tasks and put the tasks for subordinate units. Simulation objects and their methods are managed by a dedicated simulation kernel (extension of ADA language). Object methods are divided into two sets:

(1) non-simulation methods — designed in order to set and get attributes values, specific calculations and database operations;

(2) simulation methods — prepared in order to synchronous (“waitfor” methods) and asynchronous (“tell” methods) data sending.

The presented methods and their implementations are very promising in the context of Computer Assisted Exercises management and effectiveness. In our opinion we can save a lot of time and training audience, so even very complex exercise we can organise and realise due to analysis and gaming different scenario of military conflicts.

Received April 10 2008, revised November 2008.

REFERENCES

[1] R. Antkiewicz, A. Najgebauer, Z. Tarapata, *The decision automata for command and control simulation on the tactical level*, Proceedings of The NATO, Regional Conference on Military Communication and Information Systems, 07-10 October, Zegrze (Poland), 2003, (CD publication).

[2] R. Antkiewicz, W. Kulas, A. Najgebauer, D. Pierzchała, J. Rulka, Z. Tarapata, R. Wantoch-Rekowski, *Case-Based C^2 Modelling and Effective Development, Implementation and Experimentation for Simulation Based Operational Training Support System*. MSG-035, Symposium on The Effectiveness of Modelling and Simulation — From Anecdotal to Substantive Evidence, NATO RTO Publications, Warsaw, 2005.

[3] H. Bunke, *Graph matching: Theoretical foundations, algorithms, and applications*, Proceedings Vision Interface 2000, Montreal, 2000, 82-88.

[4] V. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren, *A measure of similarity between graph vertices: applications to synonym extraction and web searching*, SIAM Review, v. 46, 4, 2004, 647-666.

[5] Z. Tarapata, *Multicriteria weighted graph similarity and its application for decision situation pattern matching problem*, Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics, 27-30 August 2007, Szczecin, Poland, 1149-1155.

R. ANTKIEWICZ, W. KULAS, A. NAJGEBAUER, D. PIERZCHAŁA, J. RULKA, Z. TARAPATA, R.WANTOCH-REKOWSKI

Modelowanie i symulacja procesów C^2 bazujących na przypadkach w systemie symulacyjnym dla CAX

Streszczenie. System Symulacyjnego Wspomagania Szkolenia Operacyjnego Wojsk (SSWSO), który został skonstruowany w celu prowadzenia niskokosztowych ćwiczeń wspomaganych komputerowo (ang. Computer Assisted Exercises, CAX) jest zintegrowanym, interaktywnym, wieloszczeblowym, wspomagającym ćwiczenia i analizy postsymulacyjne systemem (z uwzględnieniem logistyki, wojsk inżynieryjnych, walki elektronicznej i rozpoznania).
W pracy przedstawiono ideę i model procesu dowodzenia i kontroli w zastosowaniu do automatu decyzyjnego na szczeblu taktycznym. Automat realizuje dwa główne procesy: proces planowania decyzji i bezpośredniej kontroli walki. Proces planowania decyzji składa się z trzech etapów: identyfikacji sytuacji decyzyjnej, generowania wariantów decyzji (planów działań), oceny wariantów i wyboru wariantu najlepszego, który spełnia pewne kryteria. Zaprezentowano specyficzne podejście do problemu identyfikacji sytuacji decyzyjnych oraz wariantów działań. Procedura generowania wariantów bazuje na procesie presymulacji i zawiera moduł oceniający, który umożliwia nam wybór najlepszego planu działań w zależności od przyjętych kryteriów. Proces bieżącej kontroli walki zawiera takie fazy, jak: dowodzenie, meldowanie i reakcję na tzw. sytuacje awaryjne. W pracy przedstawiono wyniki wybranych symulacji uwzględniających decyzje podejmowane przez opisywany automat decyzyjny.

Słowa kluczowe: modelowanie i symulacja pola walki, automat decyzyjny, system dla CAX
Symbole UKD: 355.077