Case-Based C2 Modelling and Effective Development, Implementation and Experimentation for Simulation Based Operational Training Support System

Andrzej Najgebauer PhD, DSc
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
anajgeb@isi.wat.waw.pl

Ryszard Antkiewicz, PhD, DSc
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
rantkiew@isi.wat.waw.pl

Zbigniew Tarapata, PhD
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
ztarap@isi.wat.waw.pl

Jarosław Rulka, PhD
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
jrulka@isi.wat.waw.pl

Dariusz Pierzchała, PhD
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
darp@isi.wat.edu.pl

Wojciech Kulas, PhD
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
wkulas@isi.wat.waw.pl

Roman Wantoch Rekowski, PhD
Military University of Technology,
Faculty of Cybernetics
00-908 Warsaw, Poland
rekowski@isi.wat.waw.pl

ABSTRACT

In this paper we will describe the concept for modelling and effective development, implementation and experimentation for simulation based operational training support system. 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: the identification of a decision situation, the generation of decision variants (action plans), the variants evaluation and nomination 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 phase like command, reporting and reaction to fault situations.

Najgebauer, A.; Antkiewicz, R.; Tarapata, Z.; Rulka, J.; Pierzchala, D.; Kulas, W.; Wantoch Rekowski, R. (2005) Case-Based C2 Modelling and Effective Development, Implementation and Experimentation for Simulation Based Operational Training Support System. In *The Effectiveness of Modelling and Simulation – From Anecdotal to Substantive Evidence* (pp. 6-1 – 6-16). Meeting Proceedings RTO-MP-MSG-035, Paper 6. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

1.0 INTRODUCTION

The idea and model of command and control process applied for the decision automata on the tactical level are considered.

The many "automatic commander" modules that represent different positions in the command structure of own and opposite forces of war game is considered in the Simulation Based Operational Training Support System (SBOTSS) in order to provide cost-effective approach of Computer Assisted Exercises. The idea of military unit structure used in SBOTSS is presented on the Figure 1.

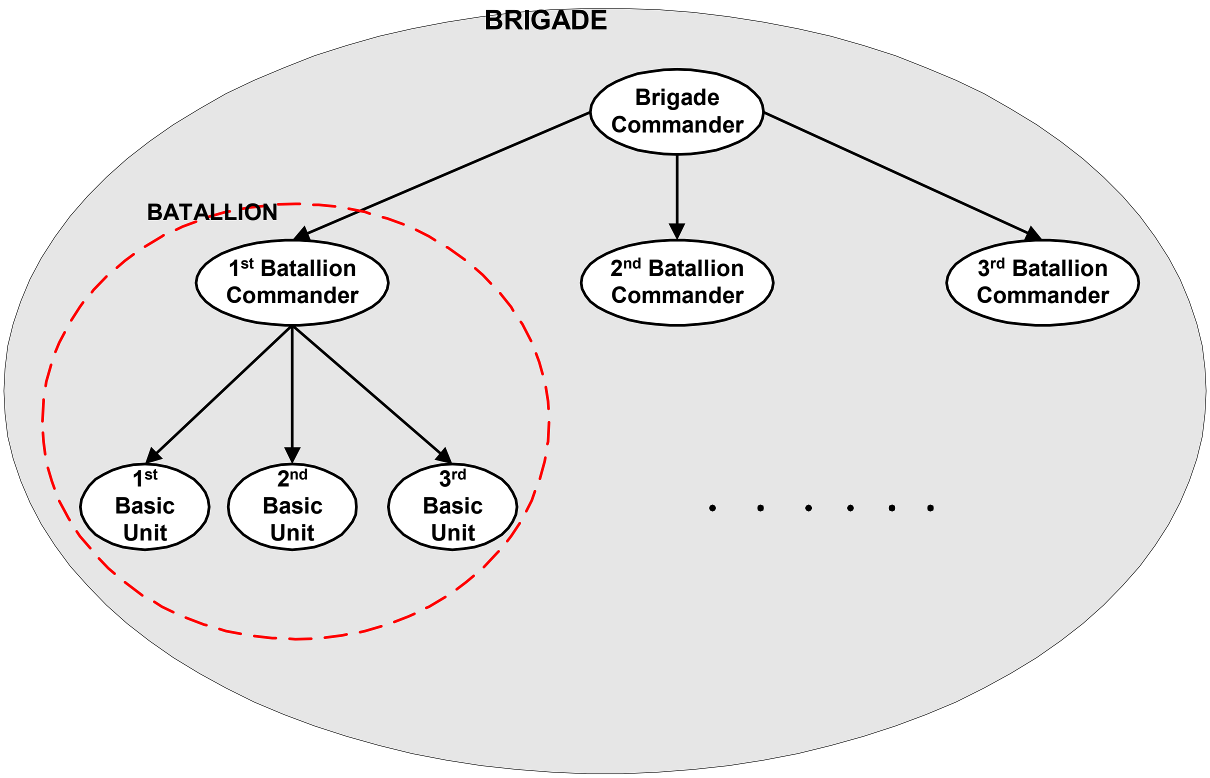


Figure 1: The structure of simulation model of Brigade.

Physically simulated objects in this structure are command posts, automatic commanders and basic units (in the SBOTSS basic units means company, battery or logistic platoon). The source of effectiveness is in the limitation of staff personnel, required in the CAXs and replacement the staff by "automatic commanders".

The automata execute two main processes: decision planning process and direct combat control. The decision planning process relating to the automata contains three stages: the identification of a decision situation, the generation of decision variants, the variants evaluation and nomination the best variant of these, which satisfy the proposed criteria (see Figure 2). 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 are 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.

We consider in the evaluation process the following criteria: time and degree of task realization, own losses, utilization of munitions and petrol.

2.0 THE MODEL OF A DECISION SITUATION

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

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

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

SD_1 - commanding level of opposite forces,

SD_2 - type of task of opposite forces (e.g. attack, defence),

SD_3 - commanding level of opposite forces,

SD_4 - type of task of own forces (e.g. attack, defence),

SD_5 - net of squares as a model of activities (interest) area

$$SD_5 = \left[SD_{ij}^5\right]_{\substack{i=1,..,SD_7 \\ j=1,..,SD_8}}, \tag{2}$$

SD_7 - the width of an activities (interest) area (number of squares),

SD_8 - the depth of an activities (interest) area (number of squares),

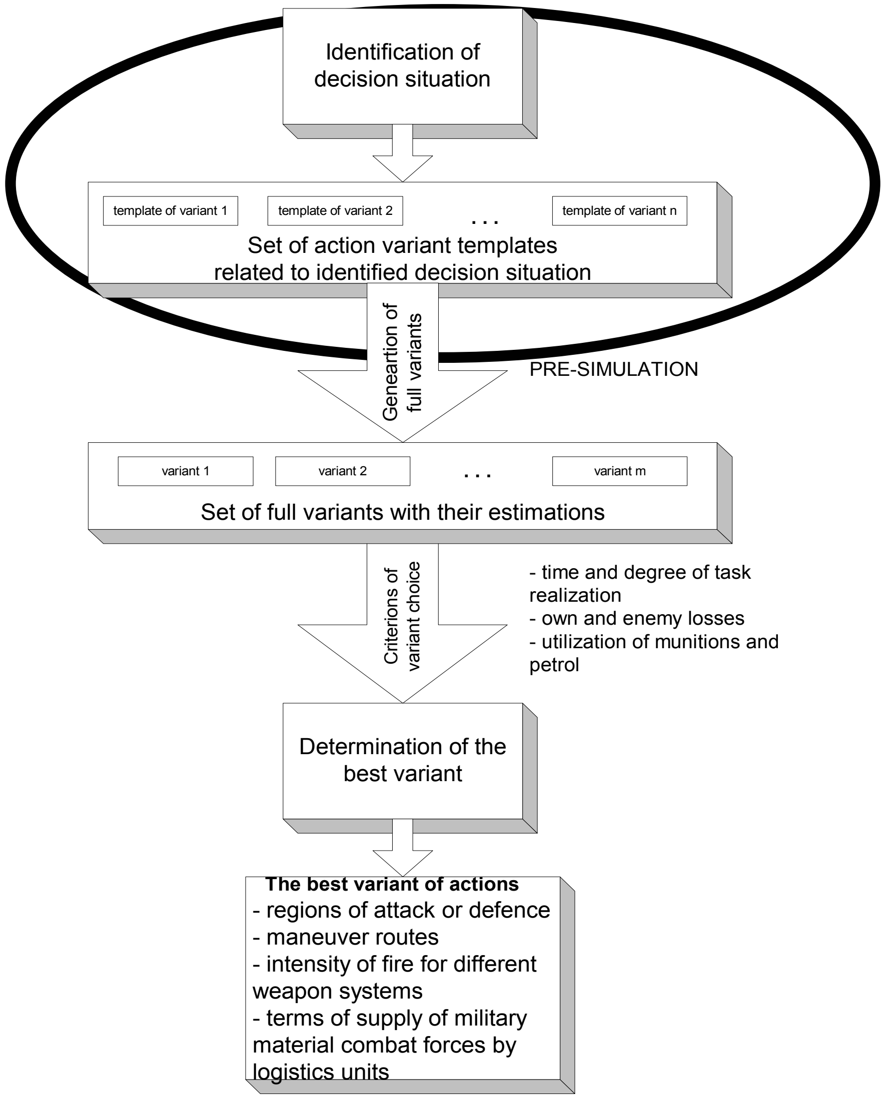


Figure 2: Algorithm for selecting the best variant of action [1].

$$SD_{ij}^{5} = (SD_{ij}^{5,k})_{k=1,..,6} \tag{3}$$

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,

$SD_{ij}^{5,4}$ - armoured power (potential) of opposite units deployed in the square,

$SD_{ij}^{5,5}$ - infantry power (potential) of opposite units deployed in the square,

$SD_{ij}^{5,6}$ - 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 - summary armoured power (potential) of own units,

SD_2^6 - summary infantry power (potential) of own units,

SD_3^6 - summary artillery power (potential) of own units,

SD_4^6 - summary air fire support power (potential).

3.0 THE FORMULATION OF THE PATTERN IDENTIFICATION PROBLEM FOR DECISION SITUATIONS

We have the set of decision situations patterns: $PDSS = \{PS : PS \in DSS\}$. For current decision situation $CS \in DSS$ we have to find the most similar situation from the set of patterns. Using the similarity measure function (4) 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 deal with following measures: distance vectors measure, weighted graphs similarity measure.

3.1. Distance vector approach

We determine the subset of 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:

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

where:

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

ΔPot - the maximum difference of own forces potential.

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

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

where:

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

$$F_{CS}\left(PS\right) = (dist_{ter}(CS,PS), dist_{pot}(CS,PS)) \tag{4}$$

$$dist_{ter}(CS,PS) = \sum_{k=1}^{3} \lambda_k \cdot \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \left(CS_{ij}^{5,k} - PS_{ij}^{5,k} \right)^p \right)^{\frac{1}{p}}, \quad \sum_{k=1}^{3} \lambda_k = 1,\ \lambda_k > 0,\ k = 1,..,3,$$

$$dist_{pot}(CS,PS) = \sum_{k=4}^{6} \mu_k \cdot \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \left(CS_{ij}^{5,k} - PS_{ij}^{5,k} \right)^p \right)^{\frac{1}{p}}, \quad \sum_{k=4}^{6} \mu_k = 1,\ \mu_k > 0,\ k = 4,..,6,$$

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

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

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

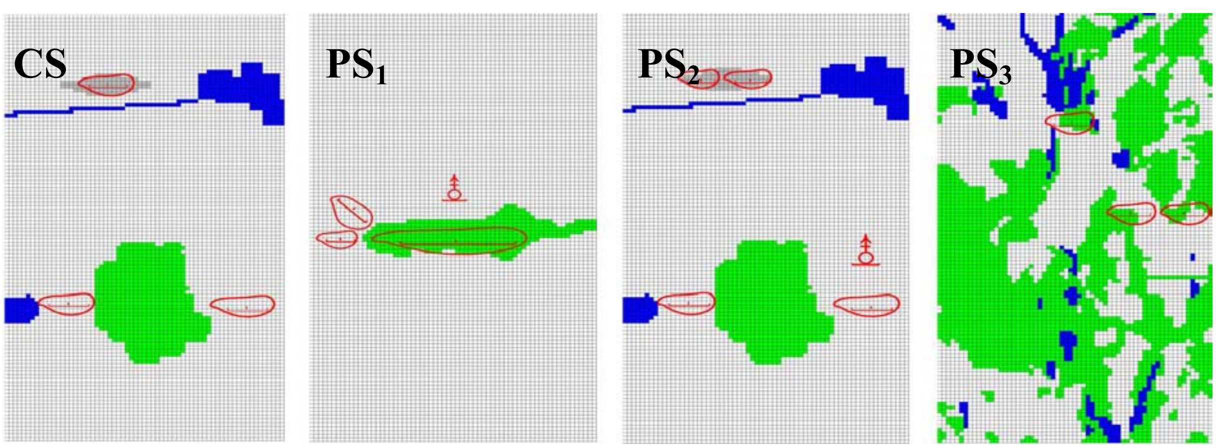


Figure 3: Hypothetical current situation CS and pattern situations (PS_1, PS_2, PS_3).

3.2. Weighted graphs similarity approach

In the literature there are several methods for determining graphs similarity (based on: graphs isomorphism[2], graphs homeomorphism [2], adjacency matrices similarity [3]). In our proposition the graphs similarity approach for identification of decision situation consists of three stages:

1. Building weighted graphs *G(CS)* and *G(PS)* representing decision situations: current (*CS*) and pattern (*PS*);

2. Calculating similarity measure *c(AS,PS)* between graphs *G(CS)* and *G(PS)*;

3. Selecting the most similar pattern situation to current situation.

Stage 1

The first stage is to build 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:

$$G = \langle V, A \rangle \tag{5}$$

where:

V – set of graph's nodes;

A – 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} from (3). On each node of G we describe some functions which identify some part of decision situation regarding considered terrain cell, e.g.: topographical conditions for graph G_T (degree of growth of : forests, waters, buildings ets., similar to: $SD_{ij}^{5,1}$, $SD_{ij}^{5,2}$, $SD_{ij}^{5,3}$), units deploying for graph G_D (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 (the more precisely: are adjacent taking into account action direction, see Figure 4). For example on the Figure 4 we have terrain divided on the 15 cells (3 rows and 5 columns). In some cells we have units (denoted by circles on the left-hand side). Structural representation of units deploying is defined by the graph G_D (right-hand side).

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

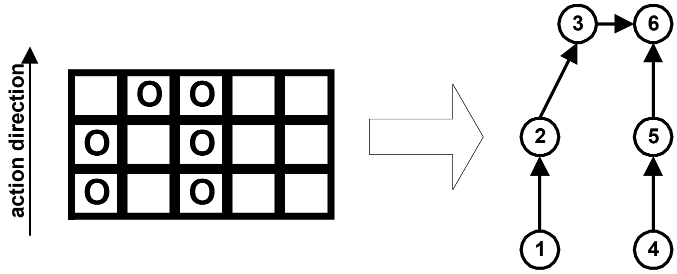


Figure 4: Units deploying and their structural (graph) representation.

Stage 2

Having weighted graphs $G_D(CS)$ and $G_D(PS)$ representing current and pattern decision situations (units deploying layer) we can modify graphs similarity approach [2], [3] to find the most similar decision situation pattern to current situation (for pair of 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 [3].

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 [3].

Let C and P define transition matrix of nodes for graph $G_D(CS)$ and $G_D(PS)$. We calculate following sequence of matrices:

$$Z_{k+1} = \frac{PZ_kC^T + P^TZ_kC}{\left\| PZ_kC^T + P^TZ_kC \right\|_F}, \quad k > 0 \tag{6}$$

where:

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

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

$$S^1 = \lim_{k \to \infty} Z_{2k} \tag{7}$$

The similarity described by elements of matrix S^1 is called "structural similarity". Element s_{ij}^1 of matrix S^1 define normalized measure of similarity between the i-th node of $G_D(CS)$ and the j-th node of $G_D(PS)$. The greater value of s_{ij}^1 the greater similarity between the i-th node of $G_D(CS)$ and the j-th node of $G_D(PS)$. 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 $G_D(CS)$ and $G_D(PS)$ we calculate distance matrices S^2 and S^3 between nodes of $G_D(CS)$ and $G_D(PS)$ from the point of view of units locations (S^2) and units military power (S^3) (functions f_1 and f_2 described on the graph's nodes). For example element s_{ij}^2 of matrix S^2 can be calculated as follows:

$$s_{ij}^2 = \frac{s_{ij}^{2*}}{\left\| S^{2*} \right\|_F} \tag{8}$$

and

$$s_{ij}^{2*} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad S^{2*} = \left[s_{ij}^{2*} \right]_{n \times m}$$

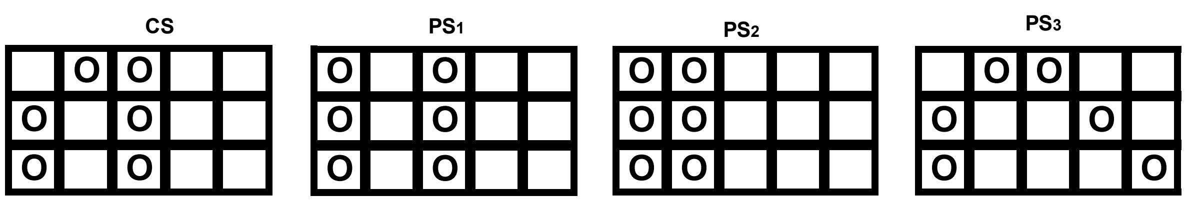
where x_i, x_j , y_i, y_j – the *x-th* and the *y-th* coordinate of the *i-th* node of the graph $G_D(CS)$ and the *j-th* node of the graph $G_D(PS)$.

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

$$s_{ij} = \sum_{k=1}^{3} s_{ij}^k \cdot \lambda_k, \quad \sum_{k=1}^{3} \lambda_k = 1, \quad \underset{k=1,\dots,3}{\forall} \lambda_k \in [0,1] \tag{9}$$

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 on the Figure 5.



0	0,11	-0,1	-0	-0,1	-0,1
-0	-0	0,12	-0,1	0,06	-0,1
-0,1	-0,1	-0,1	-0,1	-0,1	0,13
-0,1	0,02	-0,1	0,02	-0,1	-0,1
-0,1	-0,1	0,12	-0	0,12	-0
-0,1	-0,1	-0,1	0	0	0,22

$c_D(CS, PS_1)=0,256$

0,11	0,123	-0,04	0,033	-0,09	-0,09
-0,03	0,131	0,167	0,028	0,128	-0,07
-0,06	-0,06	0,111	-0,06	0,067	0,08
-0,03	0,077	0,112	-0,02	0,072	-0,07
-0,04	-0,04	0,2	-0,04	0,107	0,202
-0,07	-0,07	-0,03	-0,03	-0,03	0,191

$c_D(CS, PS_2)=0,284$

0,006	0,146	-0,03	0,075	-0,07	-0,07
-0,03	-0,02	0,194	-0,06	0,163	-0,05
-0,06	-0,06	-0,02	-0,03	-0,03	0,267
-0,09	0,045	-0,07	0,075	-0,07	-0,07
-0,08	-0,07	0,169	-0,04	0,184	-0,03
-0,07	-0,07	-0,03	0	1E-05	0,292

$c_D(CS, PS_3)=0,478$

Figure 5: Similarity matrices *S* between current decision situation *CS* and pattern situations *PS₁*, *PS₂*, *PS₃*. We set s_{ij} of *S* as $s_{ij} = s_{ij}^1 \cdot 0.5 - s_{ij}^2 \cdot 0.5$

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

$$c_D(CS, PS) = \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \cdot x_{ij} \to \max \tag{10}$$

with constraints:

$$\begin{aligned} &\sum_{i=1}^{n} x_{ij} \le 1, \quad j = \overline{1,\dots,m} \\ &\sum_{j=1}^{m} x_{ij} \le 1, \quad i = \overline{1,\dots,n} \\ &\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} = \min\{n, m\} \\ &x_{ij} \in \{0,1\} \end{aligned} \tag{11}$$

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

Stage 3

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

$$c_D(CS, PS^*) = \max_{PS \in PDSS_{CS}} c_D(CS, PS) \tag{12}$$

For example, for situations presented on the Figure 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 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.

4.0 GENERATION OF DECISION VARIANT

Having pattern decision situation most similar to current situation, we could obtain 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 set of operation plans, are evaluation of all variants of operation plan and choice the best 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:

$$Pog_B(id',t+\Delta t)=Pog_B(id',t)-\sum_{id\in JW_B^{-1}(id',t)} f\_\mathrm{int}(id,t)\cdot\Lambda_{ref}(id,t_0,id',dist(id,t,id')\cdot \cdot\frac{Pog_A(id,t)/dist(id',t,id)}{\sum_{id''\in JW_A(id.t)}(Pog_B(id'',t)/dist(id,t,id''))}\cdot\frac{Pog_A(id,t)}{Pog_A(id,t_0)}\cdot\Delta t \quad (13)$$

for $id'\in B$

$$Pog_A(id,t+\Delta t)=Pog_A(id,t)-\sum_{id'\in JW_A^{-1}(id',t)} f\_\mathrm{int}(id',t)\cdot\Lambda_{ref}(id',t_0,id,dist(id',t,id))\cdot \cdot\frac{Pog_A(id,t)/dist(id',t,id)}{\sum_{id''\in JW_B(id'.t)}(Pog_B(id'',t)/dist(id',t,id''))}\cdot\frac{Pog_B(id,t)}{Pog_B(id,t_0)}\cdot\Delta t \quad (14)$$

for $id\in A$,

where: *A, B* – sides of combat,

$Pog_A(id,t), Pog_B(id,t)$ - combat potential of two sides units.

$\Lambda_{ref}(id',t_0,id,dist(id',t,id)$ - 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 time *t*.

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

$$v_{akt}(id,t)=\min\{v_{dec},\ v_{\max}^{op}(id,t)\} \quad (15)$$

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

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

$v_{\max}(id,t)$ - maximum velocity of unit *id* depends on technical possibilities of armaments,

$StOsl\,\mathrm{Pr}\,edk$ - 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 –

$$StSBiM_A(id,t+\Delta t,k) = \max\{0,\ StSBiM_A(id,t,k) - \frac{Pog_A(id,t)}{Pog_A(id,t_0)} \cdot f\_\mathrm{int}(id,t) \cdot \cdot \sum_{i \in SO: am(i,k)=1} \sum_{j \in SO(i)} sf(id,t_0,i,j) \cdot \overline{\lambda_{ij}}(t) \cdot \Delta t\} \tag{16}$$

- utilization of petrol -

$$StMPiS(id,t+\Delta t,k) = \max\{0,\ StMPiS(id,t,k) - Z_{MPS}(id,t,\Delta t,k)\} \tag{17}$$

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

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

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

- the determination of 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 reacts to possible deviation of real trajectories in comparison to determined in planning phase.

The automata was implemented in environment of distributed interactive simulation system in ADA language and it was tested with some scenarios of land combat exercises on brigade level. The environment proposed is constructed as 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 contains two basic modules:

- *Plan_of_operation;*

- *Execute_of_Attack_Plan;*

Module *Plan_of_operation* contains the following procedures:

Ident_Dec_Sit(id)

Generate_of_Variants(id)

Pre-simulation (id)

Route_Velocity_Deter (id)

Forces_attrition_procedure (id)

Slowing_Down_Rate_Attack(id)

Utilization_of_Munitions_Petrol (id)

Choice_of_ Variant (id);

Put_Orders(id)

Situation_Report(id)

Module *Plan_of_operation* contains the following procedures:

Execute_of_Attack_Plan(id):

For all id’ $\in$ (Γ(id,t):

Execute_Task(id’),

Analysis_of_Situation_Rep(id’)

Generate_Situation_Report(id)

Modify_Plan_Attack(id).

The methods were implemented and were tested for different scenario. The automata realises its task and put the tasks for subordinate units. Simulation objects and their methods are managed by 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.

Simulation kernel is object package based upon permanent process (low level Ada language task). Simulation event are stored in one of data structures: linked list (O(n) complexity) or effective BST tree ($\log_2(n)$ complexity). Events are sorted in chronological order resulted from timestamps. The Figure 6 and Figure 7 show general diagram of simulation kernel and other important associated objects.

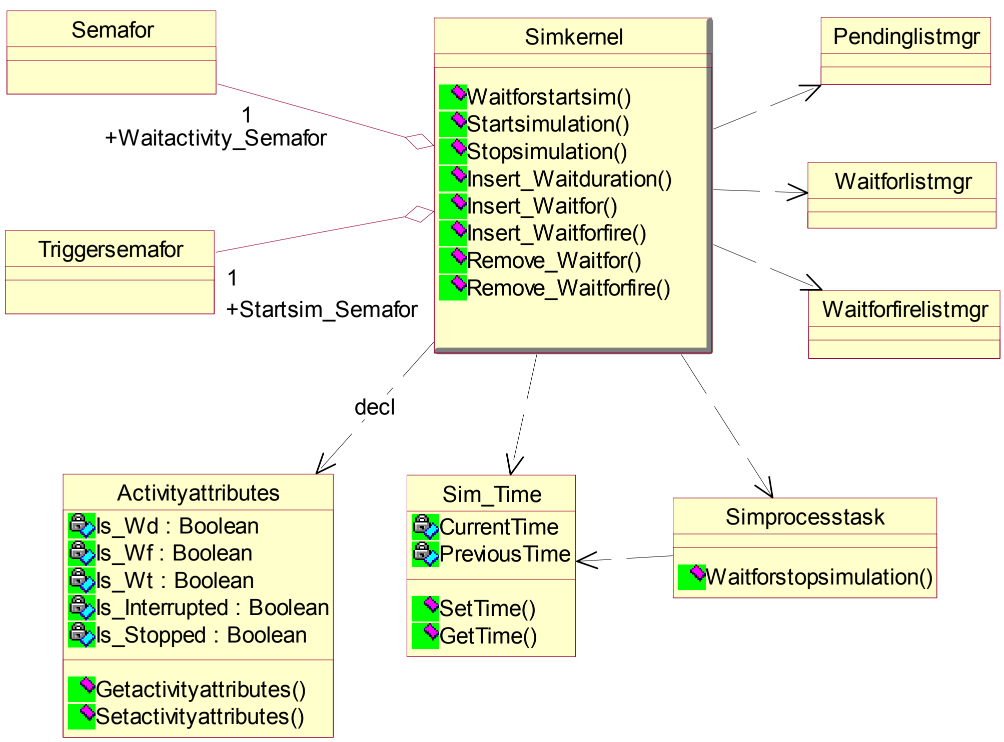


Figure 6: Class diagrams of simulation kernel package.

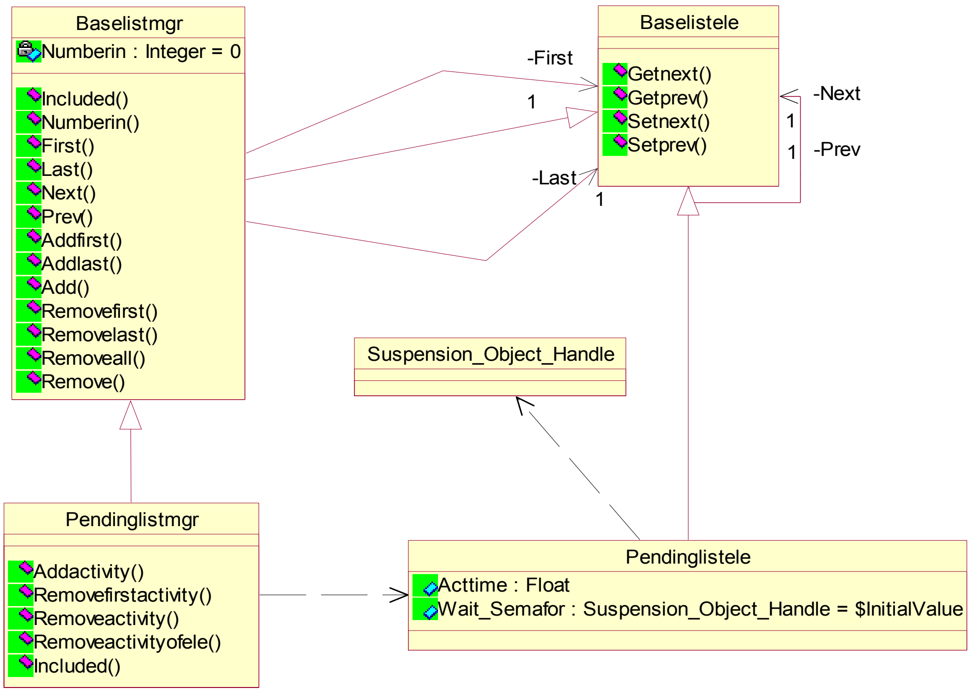


Figure 7: Diagram of classes associated with simulation kernel package.

6.0 CONCLUSIONS

Applying three approaches: the mathematical modelling (operations research methods), the FEDEP process and RUP methodology with its tools, we obtained synergy effect. We have managed very comprehensive experiment with different configurations and scenarios. Having a prototype allows us to analyze many measures connected with external and internal characteristics of distributed simulation environment and then to use these results in designing process of professional simulation system for CAX. The idea of many "automatic commanders" allows limiting of personnel required in the process of preparation and conducting exercises.

7.0 REFERENCE

[1] Antkiewicz R., Najgebauer A., Tarapata Z.: 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-ROM publication).

[2] Bunke H.: Graph matching: Theoretical foundations, algorithms, and applications, in Proceedings Vision Interface 2000, Montreal, 2000, 82 – 88.

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