BIULETYN WAT
VOL. LVII, NR 4, 2008

Army support modelling and implementation for discrete-event combat simulation

TOMASZ GÓRSKI, TADEUSZ NOWICKI,
ROMAN WANTOCH-REKOWSKI

Military University of Technology, Cybernetisc Faculty,
2 Kaliskiego Str., 00-908 Warsaw

Abstract. Manner of army support units' modelling is presented in the article. Issues of engineering, technical and medical support are main concern of the article. Especially, fortification extension, engineering barrier building and looking for injured or dead people are activities considered in the paper. Method of modelling and designing army support activities is presented. There are shown mathematical model of engineering, technical and medical support units, and tasks realized by that kind of military troops. Furthermore, unified engineering, technical and medical unit description as well as unified way of tasks execution was depicted in the article. The unification issue is interesting due to types' variety of that kind of units.
Keywords: simulation model, modelling of army support units' activity
Universal Decimal Classification: 355.077.1

1. Introduction

Main processes located on battlefield are considered in many scientific papers. In many cases only part of combat elements are shown. The main interest of scientific research is strongly connected with fighting processes [1-3]. These processes usually do not contain very important part of battlefield which is support units' activities. Nevertheless, logistic processes are very important part of military operations. This part of military activities on battlefield is the main subject of our considerations.

Support modelling issue of engineering, technical and medical units was presented in the article. Main issues of engineering, technical and medical support, which were taken into consideration in the paper, are: fortification extension, engineering barrier building and organizing, maintenance of crossings, calculating a vector of different kinds of technical equipment numbers diagnosed, real number of technical equipment located in certain area, number of people dead and injury in particular area. Each of these elements is located in simulation environment. As a preceding element, before simulation, a mathematical model of support activities on battlefield was created. Then, the model of support units' activities in simulation model was implemented.

A method of modelling and designing such issues was presented. There were shown mathematical model of engineering, technical and medical unit, and tasks realized by that kind of military troops. Furthermore, unified engineering, technical and medical unit description and unified way of tasks execution were depicted in the article. The unification issue is interesting due to types' variety of that kind of units.

There were shown tasks realized by engineering, technical and medical units (other kinds of units which are considered in battlefield) and issue of their cooperation in combat tasks execution, as well. Worth mentioning is tasks execution method which takes into consideration various conditions (e.g. influence of weather conditions).

Moreover, manner of using Unified Modelling Language (UML) to model engineering, technical and medical units and tasks were depicted as well. Presented models include class, activity, sequence and communication diagrams which were created in IBM Rational Software Architect — tool which supports software modelling, designing and developing process.

Algorithms which realize engineering, technical and medical tasks were implemented in Ada language.

It is worth to establish that a support unit in this article will be understood as engineering, technical, and medical unit.

2. Method of model construction of support units

In the following part of the article, the method of model construction of support units was presented. First of all, a mathematical model of a support unit and a task realized by this unit should be created. The steps of creating mathematical model were presented below:

- elicit and unify units which realize support tasks,
- identify and describe allocated objects,
- identify and describe positioned objects,
- identify needs,
- calculate theoretical abilities of support units,
- compute theoretical abilities of combat units,
- compute conditions of task's execution,

- compute real abilities of support units,
- describe algorithm of task's execution.

Secondly, we have to design software to execute simulation of that kind of units. We can use class diagrams of UML to describe a static structure of software and to present elements such as: unit, unit's tasks, weapons and military equipment, war and material means. Also, we can apply activity, sequence and communication diagrams to represent unit's behaviour during task's execution.

Support units can realize one of three types of tasks:

- increase in combat unit's abilities in task's execution (e.g. fortification extension),
- realize tasks which enable combat units' manoeuvres (e.g. organizing and maintenance of crossings),
- decrease in enemy combat unit's abilities in task's execution (e.g. engineering barrier building).

In the following part of the article, an engineering unit, one of support units, was described.

3. Engineering support

Engineering support it is forces' activities which consist in terrain adaptation to leading tactical operations. The aim of engineering support is creation of conditions which enable effective operating of a combat unit. It encompasses realizing engineering tasks by engineering and combat units as well.

Engineering support consists of the following tasks:

- fortification extension,
- engineering barrier building and crossings' making in minefields,
- organizing and maintenance of crossings,
- water extracting and purification,
- engineering ventures within the confines of camouflage of forces,
- preparation and maintenance of roads,
- engineering reconnaissance of enemy and terrain.

In this article only the first three tasks are considered because those tasks have the most significant influence on a situation on battlefield.

Fortification extension encompasses preparation of fortified position, which helps forces to defend their positions and decreases a number of casualties. Engineering barrier building is organized due to reinforce defence areas and makes attack difficult for enemy to conduct.

Crossings' making in minefields consists in removing mines and charges from a specified place, which helps forces to go through a minefield, and continuation of the previously planned manoeuvre.

Organizing and maintenance of crossings consists in creating right technical conditions on water obstacle which makes it possible for forces to get to the other side.

In this part of the article, a method of model construction of engineering unit was presented.

Elicit and unify units which realize engineering tasks

In the model, unified description of unit's state was established. Unit's state was described by the state vector shown below:

$$\text{Stan\_JP}_A(id,t) = \begin{pmatrix} ST_A(id,t), ZD^c_A(id,t), D^c_A(id,t), \overline{X}_A(id,t), V_A(id,t), SO_A(id,t), SW_A(id,t), \\ SM_A(id,t), StUiSW_A(id,t),\ StSBiM_A(id,t), CzasW_A(id,t), OBR_A(id,t), \\ CNR_A(id,t), OPW_A(id,t) \end{pmatrix}$$

In that vector there are elements such as number of soldiers, currently realized task, number of weapons and military equipment of each kind, number of war and material means of each type. Each unit is identified by the unique *id*.

As previously shown there are various engineering tasks realized by engineering units, which are of specific kind and type *id*(*rji*, *tji*).

The abbreviation *rji* stands for a kind of the engineering unit, $rji \in RJI$.

The set *RJI* = {1 — engineering barrier building, 2 — fortification extension, 3 — preparation and maintenance of roads, 4 — engineering reconnaissance, 5 — organizing and maintenance of crossings, 6 — water extracting and purification, 7 — camouflage of forces}.

Within each kind of engineering unit there are types of that kind of unit, $tji \in TJI(rji)$. For the first kind of engineering units set of its types is as follows, *TJI*(1) = {1 — sappers, 2 — mining, 3 — special mining, 4 — destructions, 5 — clear mines}.

An engineering machines platoon, *id*(2,1) realizes the tasks of fortification extension. For this unit, the task of support of fortification extension of the combat unit, $Rodz\_J_A(id\_wsp)$, can be set, where *id_wsp* stands for the combat unit.

Identify and describe engineering objects

Engineering units can create two types of objects:

- allocated — by distinguishing part of them,
- positioned — by creating new objects in terrain as bridges, fortified positions, minefields.

Each object created by an engineering unit can be one of the specified types $OBI(toi)$, $toi \in TOI$.

Set TOI = {1 — engineering machines' team, 2 — minefield, 3 — object prepared to destroy, 4 — pontoon bridge, 5 — pontoon ferry, 6 — landing crossing, 7 — tank crossing under water, 8 — water point, 9 — cross-country road, 10 — apparent command post, 11 — apparent section of road, 12 — apparent bridge, 13 — engineering reconnaissance group}.

Allocated objects

Engineering machines platoon can allocate engineering machines team $OBI(1)$ as one or several machines to ground works.

Positioned objects

Engineering machines platoon realizes works which change a state of a terrain occupied by a supported combat unit. In that area, the engineering unit changes the state of fortification extension. The state of fortification extension $SRF(t)$ in a square $w_1 \in W_1$ is described as follows:

$$SRF(t) = (srf\_nr(t), srf\_p(t))$$

$srf\_nr(t)$ is the number of latest realized sequence of fortification works, ($srf\_nr \in N$),

$srf\_p(t)$ is the percent of realized fortification works of the sequence $srf\_nr(t)$, ($srf\_p \in [0,100]$).

Identify needs

Supported combat units have different needs for fortification extension. Needs for fortification extension depend on the organizational level $Szczeb\_J_A(id)$ and the kind of unit $Rodz\_J_A(id)$.

It was assumed in the model that an amount of fortification extension needs of the sequence k $PRF(k, Szczeb\_J_A(id), Rodz\_J_A(id))$ depends on the organizational level $Szczeb\_J_A(id)$ and the kind of unit $Rodz\_J_A(id)$.

For example, fortification extension needs for defence area of mechanized battalion are as follows:

— volume of ground (m^3) = 11200,
— all needs (ms) = 1388,
— needs of first sequence (ms) = 354.

It was assumed that one day of work of one soldier (ms) is equal to 10 hours of work of one soldier (mh) because 1 day and night = 10 h. So, an amount of work to do is calculated in days of work of one soldier.

Theoretical abilities of unit

Abilities of the fortification extension $MRF^{c}(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))$ unit of organizational level $Szczeb\_J_A(id)$ of the kind $Rodz\_J_A(id)$ are a sum of abilities of fortification extension of this unit $MRF(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))$ and abilities of assigned fortification extension units.

$$MRF^{c}(Szczeb\_J_{A}(id), Rodz\_J_{A}(id)) =$$
$$= \left(\sum_{i=1}^{N} MRF(przy\_id)\cdot m(id,i)\right) + MRF(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))$$

where: $m(id,i)$ — the number of fortification extension units of the level i assigned to the unit *id*;
$MRF(przy\_id)$ — the fortification extension abilities of the unit *przy_id* assigned to the unit *id*.

Theoretical abilities of combat unit

Abilities of the fortification extension $MRF(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))$ of the unit *id* can be expressed as product:

$$MRF(Szczeb\_J_{A}(id), Rodz\_J_{A}(id)) =$$
$$= PSO(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))\cdot so_{A}(id,t,1)$$

where: $so_{A}(id,t,1)$ — the number of soldiers in the unit *id* at the moment *t*;
$PSO(Szczeb\_J_{A}(id), Rodz\_J_{A}(id)) \in [0, 1]$ — the manpower percent of the unit organizational level $Szczeb\_J_{A}(id)$, of the kind $Rodz\_J_{A}(id)$ which can be sent to fortification extension works.

For example $PSO(Szczeb\_J_{A}(id), Rodz\_J_{A}(id))$ for mechanized and infantry detachments is 0.7.

Theoretical abilities of engineering unit

Abilities $MRF(przy\_id)$ of the fortification extension unit result from number of possessed machines to ground works of each type, $tpmz \in TPMZ$.

$$MRF(przy\_id) = \sum_{k=1}^{K} w_{k}\cdot l_{k}(t),$$

where: $k = \overline{1,K}$ — stands for the number of a type of machine to ground works;
w_k is the machine to ground works' type k performance expressed in rbh;
$l_k(t, przy\_id)$ is the number of machines to ground works in working order, of the type k in instant t in unit *przy_id*.

For example, TPMZ(1)=KRS and its performance is 270 [man-hour].

Count real abilities

Each task on battlefield is executed in real conditions so, the described earlier abilities are only theoretical ones. In order to obtain real abilities of task execution we have to take into consideration weather and topographical conditions.

Modelling conditions of task execution

The symbol $\Psi_{1,5}(\cdot,t)$ represents function, which defines topographical conditions.

$$\Psi_{1,5}(\cdot,t) = (KTG(\cdot,t), RZT(\cdot,t), STZ(\cdot,t), WTP(\cdot,t), RDD(\cdot,t), RCD(\cdot,t), PPW(\cdot,t)),$$

where: $KTG(\cdot,t)$ is the ground category, $KTG(\cdot,t) \in$ {1 — loose, 2 — average hardness, 3 — tough, 4 — rocky};
$RZT(\cdot,t)$ — the relief, $RZT(\cdot,t) \in$ {1 — plain, 2 — hilly, 3 — mountainous, 4 — desert, 5 — marshy};
$STZ(\cdot,t)$ — the afforest ration, $STZ(\cdot,t) \in$ {1-<= 25%, 2 — (25%, 50%>, 3 — (50%, 75%>, 4 — > 75%}.

The symbol $\Psi_{1,1}(\cdot,t)$ represents a function that defines weather conditions.

$$\Psi_{1,1}(\cdot,t) = (GMZ(\cdot,t), IOA(\cdot,t), PDN(\cdot,t), PRK(\cdot,t)),$$

where: $GMZ(\cdot,t) \in R^+$ [meter] is the depth of frozen ground in the area $\overline{X}_A(id,t)$;
$IOA(\cdot,t)$ is the intensity of precipitation in the area $\overline{X}_A(id,t)$;
$IOA(\cdot,t) \in$ {1 — to 2,5 [mm/h], 2 — from 2,5 to 13 [mm/h], 3 — above 13 [mm/h]};
$PDN(\cdot,t)$ — day or night in the area $\overline{X}_A(id,t)$, $PDN(\cdot,t) \in$ {1 — day, 2 — night};
$PRK(\cdot,t)$ — season, $PRK(\cdot,t) \in$ {1 — spring, 2 — summer, 3 — autumn, 4 — winter}.

Applying conditions to abilities

Real abilities of fortification extension depend on terrain characteristics and weather conditions. Below, were shown several of identified factors, expressed by coefficients, which have impact on soldiers' and machines' work:

ground category — $K_g^s \in <0, 1>$,
relief — $K_t^s \in <0, 1>$,
afforest ration — $K_l^s \in <0, 1>$,
depth of frozen ground — $K_z^s \in <0, 1>$,

where: s — the number of type of work execution, $s \in SWP$, SWP={1 — manually, 2 — mechanical};
Values of each mentioned above coefficient are tabularised quantities.

In order to get real fortification extension abilities of unit you have to apply product of the above identified coefficients:

$$\mathrm{K}^{s} = \mathrm{K}_{g}^{s} \cdot \mathrm{K}_{t}^{s} \cdot \mathrm{K}_{l}^{s} \cdot \mathrm{K}_{p}^{s} \cdot \mathrm{K}_{z}^{s} \cdot \mathrm{K}_{a}^{s} \cdot \mathrm{K}_{w}^{s} .$$

Real abilities of fortification extension can be expressed by Eq. (1) depicted below:

$$MRF_{rzecz}^{c}(Szczeb\_J_{A}(id), Rodz\_J_{A}(id)) = \sum_{s=1}^{SWP} MRF_{s}^{c}(Szczeb\_J_{A}(id), Rodz\_J_{A}(id)) \cdot \mathrm{K}^{s} . \qquad \text{Eq. (1)}$$

Describe algorithm of task execution

It is important to elicit a standard task model. Fortification extension task of the unit *id_wsp* can be set for the engineering machines platoon *id*(2,1) and the object *OBI*(1).

The fortification extension task is as follows:

$$ZD_{A}^{c}(id,t) = \left(z, id\_wsp, r, t^{p}, t^{k}, r^{z}(id)\right)$$

z — the kind of task, $z \in$ {support, allocation},
id_wsp — supported unit,
r — the layout area of unit *id_wsp*, $r \in W_1$,
t^{p} — the starting moment of fortification extension (the time of readiness fire system t_{GSO}),
t^{k} — the moment, to which the unit *id* is to support the unit *id_wsp* in fortification extension,
$r^{z}(id)$ — the marshalling area after task execution.

To realize such a task, it is essential to define algorithms realized by the unit *id*, $alg \in ALG$ = {1 — determine route of march, 2 — execute march, 3 — execute fortification extension}.

Algorithm of fortification extension

Input data:

t_{max} — the maximal time of uninterrupted work of soldiers and machines in day and night,

Event activation or the starting moment: t_{GSO} — the readiness time of a fire system of the unit *id_wsp*.

Algorithm body:

Step-1 — calculate fortification extension abilities of the unit *id_wsp* including supporting units according to Eq. (1), go to step 2,

Step-2 — start of fortification extension realisation for time 1 h. If $t = t^k$ or required $srf\_nr(t)$ is realized in $srf\_p(t) = 100$ then, the end of algorithm functioning, else go to step 3,

Step-3 — update $SRF(t)$ for $LRK(id\_wsp)$. If the time of uninterrupted work of the fortification extension $t_{nr} < t_{max}$ then go to step 2, else go to step 4.

Step-4 — suspend realisation of algorithm the of fortification extension unit *id_wsp* at the time $t_z = 24 - t_{max}$. Set $t_{nr} = 0$. Go to step 2.

4. Technical diagnosis model with technical support equipment

We assume, that:

$\hat{X}\ (id, t)$ is the area where the technical support equipment *id* is going to do technical diagnosis.

$\hat{X}\ (id, t) = (X_1(id, t), X_2(id, t), ..., X_N(id, t))$ — is the sequence of adjacent square of a terrain model.

$UiSWrozp(\hat{X}\ (id, t), id, t)$ is the function which calculates a vector of different kinds of UiSW numbers in $\hat{X}\ (id, t)$ area. The vector consists of the same number of elements as the number of different kinds of UiSW.

$Rozpoznanie(UiSWrozp(\hat{X}\ (id, t), id, t), id, t, Z1)$ — the function calculates the vector of different kinds of UiSW numbers diagnosed in $\hat{X}\ (id, t)$ area. The vector consists of the same number of elements as a number of UiSW different kinds.

The values of $Rozpoznanie(\bullet)$ function are transformed according to a real number of UiSW located in a certain area. The values can be lower because of the conditions included in Z1 i Z2 models.

$Rozpoznanie_i(\bullet) = \lfloor P_{0-1} * wp * wd * ww * UiSW_i(\bullet) \rfloor$

$\bullet$ — the number of i-th UiSW.

P_{0-1} — the random value from a range [0,1] (this value includes all random elements which can affect technical diagnosis exclude above parameters)

$wp(\bullet)$ — the weather conditions in $\hat{X}\ (id, t)$ area (season, rein, fog, smoke, snow etc.) — the values from the range [0,1], where 1 means perfect weather conditions

$wd(\bullet)$ — the travel conditions in $\hat{X}\ (id, t)$ area (mobile, lakes, rivers, mountains, etc.) — the values from the range [0,1], where 1 means perfect road conditions in $\hat{X}\ (id, t)$ area

$ww(\bullet)$ — the visibility conditions in $\hat{X}\ (id, t)$ area $\hat{X}\ (id, t)$ (trees, mountains, etc.) — the values from the range [0,1], where 1 means perfect visibility conditions in $\hat{X}\ (id, t)$ area.

Movement model with technical support equipment and military units

The path technical support equipment and military units is defined as pairs sequence of vertex and speed on arcs in Z2(t) model:

$$Dr_A(id,t) = \big(w(id,t,m), v(id,t,m)\big)_{m=1,\overline{LW(id)}}$$

$w(id,t,m)$ — the m-th vertex in path for technical support equipment and the military units *id*,

$$w(id,t,m) \in \begin{cases} W_2(t) - \text{when vetrex } w(id,t,m) \\ \qquad \text{is the first vertex in path} \\ \qquad \text{of reailroad or vehicle road} \\ W_1(t) - \text{when vetrex } w(id,t,m) \\ \qquad \text{is the first vertex in path} \\ \qquad \text{of seaway or airway} \end{cases}$$

$LW(id,t)$ — the number of vertexes in a path of technical support equipment and military units *id*,

$v(id,t,m)$ — the military units speed *id* on arc in m-th vertex.

It is necessary to fix a march column front according to the tactical norms (length of march column).

$NRw(Dr_A(id,t),t)$ — the function which calculates number of vertex where the front of march column is at t moment. The value of this function makes sense only for technical support equipment and military units.

Setting up technical support equipment model

Technical-tactical data fix the time of setting-up technical support equipment. We assume that

$TRUrz_{NORMA}(id)$ is the standard time of the setting-up technical support equipment *id*,

$\hat{X}\ (id,t)$ is the area where the technical support equipment *id* will be setting-up.

The time of setting-up the technical support equipment *id* is calculated by the function:

$$TRUrz(id,t,\hat{X}\ (id,t),TRUrz_{NORMA}(id)) =$$

$$\frac{TRUrz_{NORMA}(id)}{Ukompl(id)*wt(\hat{X}\ (id,t),t)*wp(t)*P_{0-1}},$$

where

$Ukompl(id)$ — the number of soldiers and equipment, values from the range [0,1], where 1 means full number of soldiers and equipment,

$wp(\bullet)$ — the weather conditions in $\hat{X}\ (id,t)$ area (season, rein, fog, smoke, snow etc.) — the values from the range [0,1], where 1 means perfect weather conditions,

$wt(\bullet)$ — the travel conditions in $\hat{X}\ (id,t)$ area (mobile, lakes, rivers, mountains, etc.) — the values from the range [0,1], where 1 means perfect road conditions in $\hat{X}\ (id,t)$ area,

$P_{0-1} - P_{0-1}$ — the random value from the range [0,1] (this value includes all random elements which can have effect on technical diagnosis, exclude above parameters).

UiSW classification repair model

As a result of technical diagnosis activity, the following detailed information is available: number of diagnosed equipment for each kind of UiSW $Rozpoznanie(\bullet)$. It is necessary to classify each item to certain kind of repair.

Combat losses of mechanized unit UiSW may be defined as the matrix:

$StrStrat[RodzajUiSW, I-II, Klasa]$.

That matrix consists of percentage losses for every kind of UiSW for a certain place in mechanized unit formation.

$I-II \in \{I, II\}$ is the mechanized unit formation, where technical diagnosis, repair and evacuation has been done

$Klasa \in \{PT,1,2,3,4,5,SB\}$ — kinds of losses

Each item of UiSW is classified to a certain kind of repair.

Function $Krem(Rozpoznanie_i(\bullet), I-II)$ calculates a vector which consists of 7 numbers where:

$Krem_1(Rozpoznanie_i(\bullet), I-II)$ — the number of *i*-th UiSW kind for simple repair,

$Krem_2(Rozpoznanie_i(\bullet), I-II)$ — the number of *i*-th UiSW kind for repair (1 level of repair),

$Krem_3(Rozpoznanie_i(\bullet), I-II)$ — the number of i-th UiSW kind for repair (2 level of repair), and so on up to level 5.

$Krem_7(Rozpoznanie_i(\bullet), I-II)$ — the number of i-th UiSW kind not for repair.

The values for each kind of repair are calculated as follows: experts calculate a value of p parameter. For example, $p = 0.05$. Six values: p_1, p_2, p_3, p_4, p_5, p_6, are fixed on the base of p value as follow:

$$p_1 = 2 * p * P_{0-1} - p$$

similarly, we do the same for other parameters p_2, p_3, p_4, p_5, p_6, where:

$P_{0\text{-}1}$ — Random value from the range [0,1].

$$\begin{aligned}&Krem_1(Rozpoznanie_i(\bullet), I-II) =\\&Rozpoznanie_i(\bullet) * StrStrat[RodzajUiSW, I-II, PT] +\\&StrStrat[RodzajUiSW, I-II, PT] * p_1\end{aligned}$$

$$\begin{aligned}&Krem_2(Rozpoznanie_i(\bullet), I-II) =\\&Rozpoznanie_i(\bullet) * StrStrat[RodzajUiSW, I-II, 1] -\\&StrStrat[RodzajUiSW, I-II, 1] * p_1 +\\&StrStrat[RodzajUiSW, I-II, 1] * p_2\end{aligned}$$

and similar other values.

In this way, an engineering task is described.

5. Medical support

The main types of activity for medical service support units should be listed. Obviously, like in every case, the model contains only fundamental and important tasks for adequate describing processes on battlefield. It is known, that it is impossible to model every kind of medical service support units activity. Thus, only the main tasks, important for adequate model, were chosen. It was assumed, that medical units can realize the following types of tasks:

A. march preparation,
B. march of a whole medical service support unit,
C. march of a medical convoy to a selected sector,
D. transportation of wounded soldiers,
E. positioning of medical object in a region of allocation,
F. giving qualified help,

G. operation against epidemic treatment,
H. hospitalization of wounded or ill soldiers,
I. burial activity,
J. rolling of medical object in a region,
K. recognition of medical situation in a region.

All tasks listed above for medical service support units can be named as elementary tasks. They are the basis for more complicated activity description. The detailed description and algorithms for elementary tasks are needed to construct complex tasks of medical service support units. Therefore, in the first order, the algorithms of elementary tasks will be described and then more complex algorithms can be commented.

In order to show the way to unify description of model elements we can analyze an example connected with a medical object located in a particular medical unit. We assume that *id* is the identifier of a medical unit which is connected with many medical objects. So that, the state of this unit for the given moment *t* can be described as a vector:

$$OBR_A(id,t) = \left(OBRM_A(id,t,j)_{j=1,..,LOR_A(id,t)} \right)$$

where $OBRM_A(id,t)$ is the selected object connected with the unit with *id* identifier,

$$OBRM_A(id,t) = \begin{pmatrix} idn, TOR_A(id,t), STO_A(id,t), ZDO_A^c(id,t), \overline{XO}_A(id,t), \\ SOO_A(id,t), StUiSWO_A(id,t),\ StSBiMO_A(id,t), PAR_A(id,t) \end{pmatrix}$$

where for example:

idn — the identifier of a medical unit connected with this object,

$TOR_A(id,t)$ — the type of an object,

$TOR_A(id,t) \in \{1,2,3,4,5\}$ — the numbers describing particular possible types of object,

$PAR_A(id,t)$ — the parameters of a medical object at the moment *t*:

$$PAR_A(id,t) = \begin{pmatrix} PRR_A(id), CR_A(id), CZ_A(id), PH_A(id), PK_A(id), PE_A(id), \\ LOH_A(id,t), LOPK_A(id,t), LOE_A(id,t) \end{pmatrix}$$

$PRRO_A(id)$ is the size of an area in which medical object, that is connected with the unit *id*, is located,

$CRO_A(id)$ — the time of object positioned by the medical unit *id*,

$CZO_A(id)$ — the time of an object rolled by the medical unit *id*,
$PH_A(id)$ — the throughput of hospitality in a medical object in the sense of a person number per a day,
$PK_A(id)$ — the throughput of qualification help in a medical object in the sense of a person number per a day,
$PE_A(id)$ — the throughput of evacuation per a day using given medical vehicles,
$LOH_A(id,t)$ — the number of a hospitalized person in a medical object at the moment *t*,
$LOPK_A(id,t)$ — the number of a person for whom qualified help is given in a medical object at the moment *t*,
$LOE_A(id,t)$ — the number of a person evacuated to an object to the moment *t* using given transportation vehicles,
$NLP_A(id)$ — the quantity of personnel in considered object,
$NLUiSW_A(id)$ — the number of medical vehicles in an object.

Only an example of a medical object of a medical unit was shown above. These parameters should show the way how to unify describing medical service support unit for computer simulation. Particular algorithms for elementary tasks realized by medical units will be presented in the next section of the article.

At the beginning of the article there were listed elementary tasks connected with medical service support activity. The examples of these algorithms are given below.

<u>Algorithm A</u>. (march preparation)
Input data:
$RDZJW = 3$ (march) given from $RDZ_A(id,t,k)$
$t_0^A(id,t,k)$ — the initiation moment of march
$swp_A(id,t,UISW90,1)$, $swp_A(id,t,UISW90,2)$, $SO_A(id,t,1)$

Algorithm body:

Step-1. If the unit *id* is ready to operation $(ST_A(id,t)=1)$, then continue algorithm, else if $(ST_A(id,t)=0)$, then statement (unit is not able to be active) is sent and stop algorithm.

Step-2. Time needed for march preparation is generated CPM, (assumption: unit *id* is rolled).

If *MAXSO* — the unit staff is full completed — number of persons,
MAXUISW — the unit medical equipment is full completed — number of cars, then

$$a = \frac{MAXUISW}{swp_A(id,t,UISW90,1) + swp_A(id,t,UISW90,2)}$$

$$b = \frac{MAXSO}{SO_A(id,t,1)}$$

where: t — the actual time,

a — the time needed for march preparation for the unit *id* is fixed as: $a \cdot b \cdot CPM$.

Step-3. If $t + a \cdot b \cdot CPM \le t_0^A(id,t,k)$, then next event is generated at the moment $t_0^A(id,t,k)$ as march beginning.

Step-4. If $t + a \cdot b \cdot CPM > t_0^A(id,t,k)$, then statement (task is not realized in a primary form) is sent and we wait for information about necessity of task continuation. If this information is given, then the next event is generated at the moment $t + a \cdot b \cdot CPM$ as march beginning.

Algorithm stop.

Algorithm B. (march of a whole medical service support unit)

Input data:

$Dr_A(id,t) = (w_A(id,t,m))_{m=\overline{1,W(id)}}$ — the route for the unit *id* given by *m* periods,

$SPM_A(id,t)$ — the average speed of the unit *id*,

$DKM(id)$ — the medical convoy length of the medical unit *id*, *t* is the actual time.

Algorithm body:

Step-1. For the route $Dr_A(id,t)$ we construct the series $Dl(id,t)$ of m arcs length between nodes of a route $DL(id,t) = (dl_A(id,t,m))_{m=\overline{1,W(id)-1}}$.

The lengths of arcs are given from the graph G_2 parameters (FRodzŁuku).

Step-2. Head of medical convoy is fixed in the following way:

assuming the fact, that the average speed is $SPM_A(id)$, we calculate: on which arc the head on convoy should be located,

we sum a length of arcs on a given route and calculate a length of distance covered by a medical unit till the time *t*: $SPM_A(id) \cdot t$,

by arcs of route elimination we fix the arc on which the head of convoy is,

proportion between a length of route covered by a convoy and a sum of given arcs define us degree of actual arc terminantion.

Step-3. We fix the end of a convoy by subtracting from the value $SPM_A(id)\cdot t$ length of the convoy $DKM(id)$.

If the value $SPM_A(id)\cdot t$ is less than $DKM(id)$, so the end of the convoy is at the beginning of a route (in node $w_A(id,t,1)$).

If the value $SPM_A(id)\cdot t$ is greater than $DKM(id)$, we obtain the difference $SPM_A(id)\cdot t - DKM(id)$ and define it as $SPM_A(id)\cdot t$ in step 2 we fix the arc, on which the end of convoy is, in the same way as in step 2.

Step-4. We accept the fact, that the unit *id* covered the distance $Dr_A(id,t)$, if the time

$$\frac{\sum_{m=1}^{W(id)-1} dl_A(id,t,m) + DKM(id)}{SPM_A(id)}$$

is over. We send information about a task terminated by the unit *id*.

Step-5. We accept the fact, that the unit *id* is the node $W_A(id,t,W(id))$. If we would like to know allocation of that unit around a node we should generate numbers of quadrates $X_A(id,t,1)$, $X_A(id,t,2)$ adequate to the unit *id* size.

Algorithm stop.

It is possible to built algorithms for more complex decision for elementary medical units using algorithms of elementary tasks. Complex decisions in this case are connected with medical units activity given to other basic military units being on battlefield. Most often this activity is related with:

- medical recognition and medical operation in a particular region,
- medical service for soldiers in medical units or in a medical object which are positioned by a medical unit.

In first case, realization of such a complex decision brings us to construct series of elementary tasks: A, B, C, D, F, G, I and k. In the second case, for example, we can construct series of elementary tasks: E, F, G, H and J.

It is easy to show that every more complicated activity of a medical unit can be replaced by series of elementary tasks (elementary algorithms) commented in this part of the article.

6. Design of support units and tasks

After creating a mathematical model of engineering, technical and medical unit and its tasks Unified Modelling Language was used to design software to simulate functioning of engineering units. In the following part, example models created in IBM Rational Software Architect were presented.

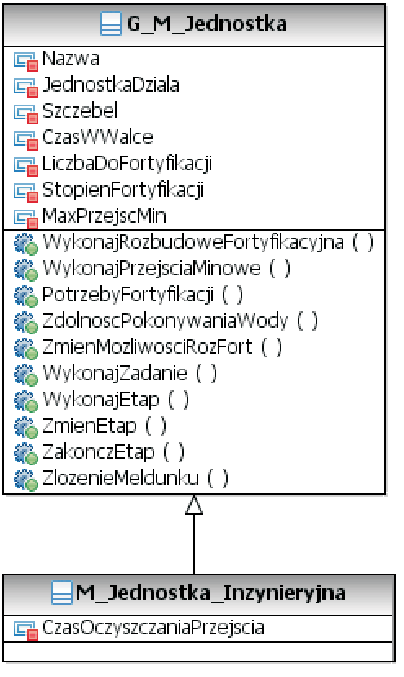
First of all, the holistic use case model was created. This model served us as a definition of a scope of system's functionality. Each use case was modelled using activity diagram which presents complete flow of events in use case.

Furthermore, a design model was created. A static structure of software was described on class diagrams and a dynamic structure of software was presented on sequence and communication diagrams.

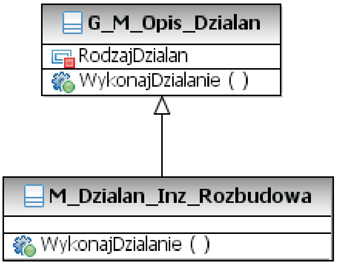
Static structure of software

On class diagrams there were depicted the classes which represent key abstractions of modelled domain, e.g. a model of engineering unit, its task, weapons and military equipment.

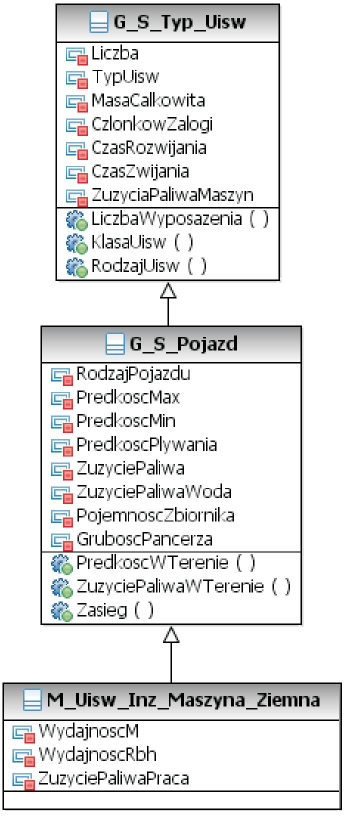
Engineering unit (M_Jednostka_Inżynieryjna) class which derives attributes and methods from the parent unit (G_M_Jednostka) class was presented on the following class diagram.



The class diagram presenting the task of engineering unit (M_Dzialan_Inz_ Rozbudowa) that derives from the general task of unit (G_M_Opis_Dzialan) was depicted on the following class diagram.



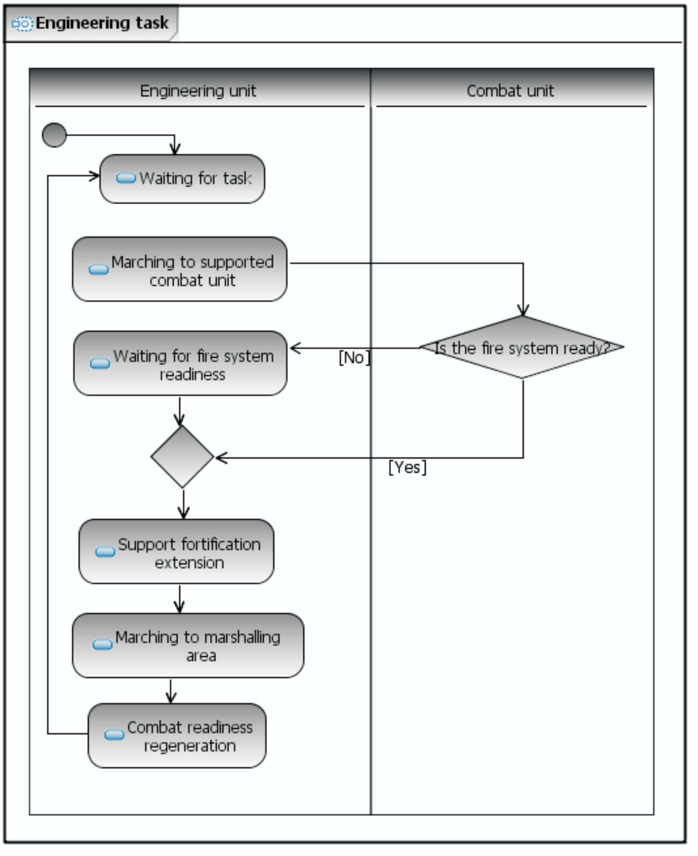
On the next class diagram, the classes with weapons' and military equipment's definition were shown.



Dynamic structure of software

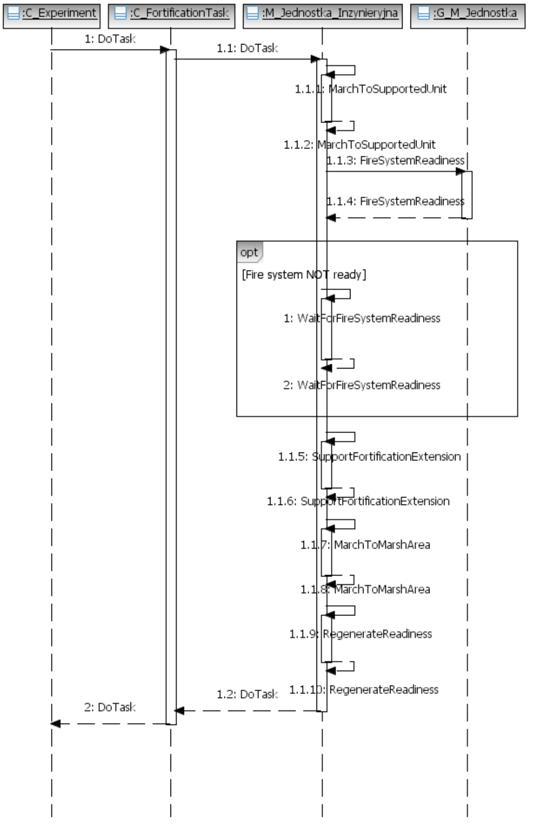
In order to identify all steps of algorithm activity, the diagrams were used.

Behaviour of the unit during executing the task of fortification extension was presented on the following activity diagram.

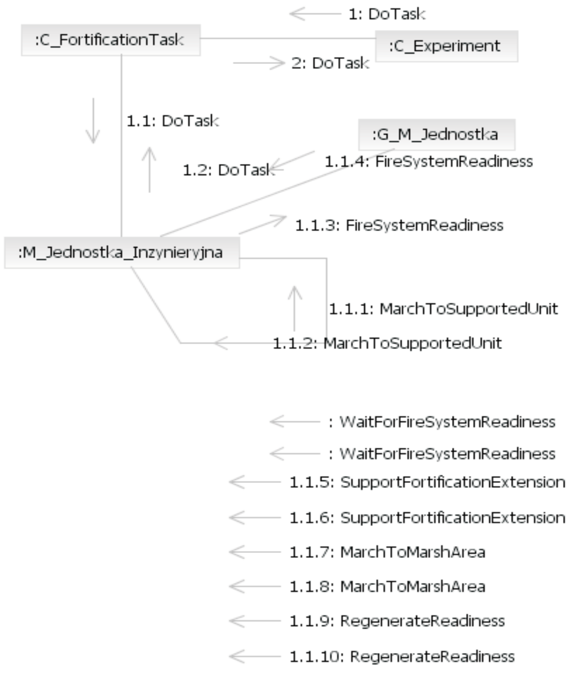


Having identified all activities we need to identify the classes and operations required to realize fortification extension task.

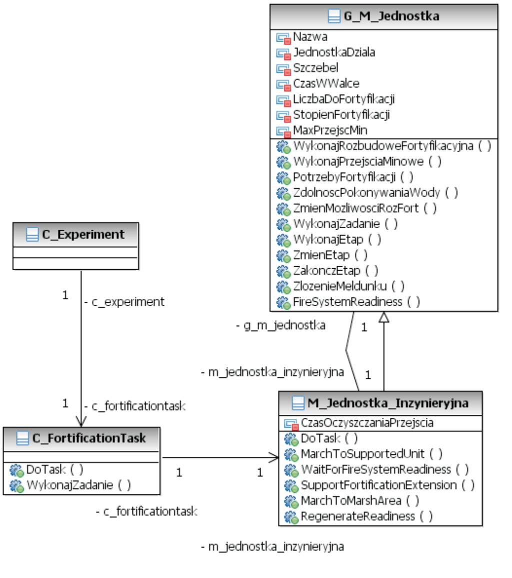
In order to do that, sequence and communication diagrams were used. An example of a sequence diagram which presents realization of fortification extension task by the designed classes is shown in the following sequence diagram.



As a next step, the communication diagrams were generated from designed sequence diagrams. Example of communication diagram which presents realization of fortification extension task by designed classes is shown in the following communication diagram.



A pattern of communications (relationships) is clearly visible on communication diagrams. The diagrams of such kind were used to refine class diagrams. On the next class diagram there were presented the relationships derived from the previous communication diagram.



Unified Modelling Language helped us in creation of a complete model of units and tasks.

Using IBM Rational Software Architect we were able to generate definitions of objects in Ada language. Part of engineering unit definition in Ada language was shown below.

```
type Object is new G_M_Jednostka.Object with
record
 Max_UnInterr_WorkTime: Float;
 UnInterr_WorkTime : Float;
 Fortification_Abilities : Types.RBH;
 Supported_Unit : G_M_Group.Handle;
 .........
 end record;
```

The next and last step was implementation of operations in refined classes.

7. Simulation of support activity

It was built simulation environment for polish army decision makers training.

The palette for modelling elements of battlefield (military units, logistics, medical units, etc.) is shown below (Fig. 1).

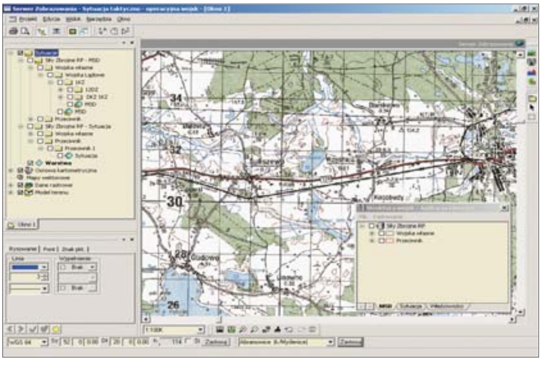


Fig. 1. Palette for battlefield modeling

We can observe tactical situation during the battle (Fig. 2) even with many elements connected with support tasks: logistics, medical, technical, etc.

We can describe tasks, for example, connected with technical evacuation of UiSW (Fig. 3) or parameters for a medical object (Fig. 4).

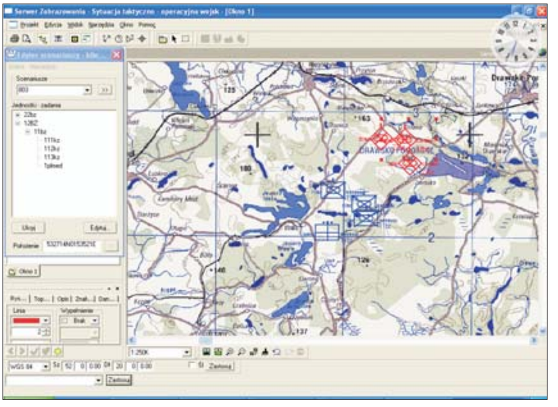


Fig. 2. Tactical situation with information of support units activity

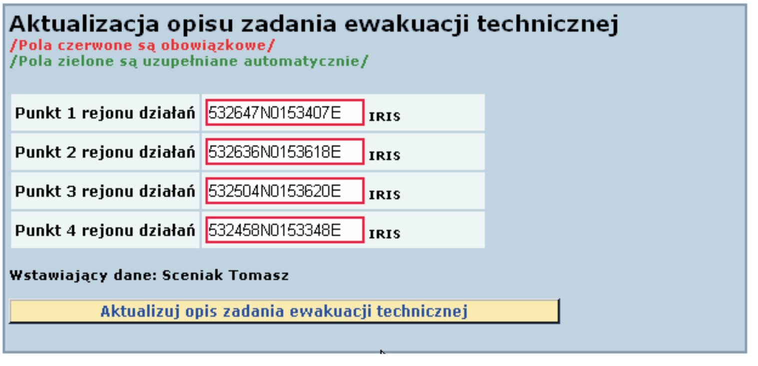


Fig. 3. Example of support units interface

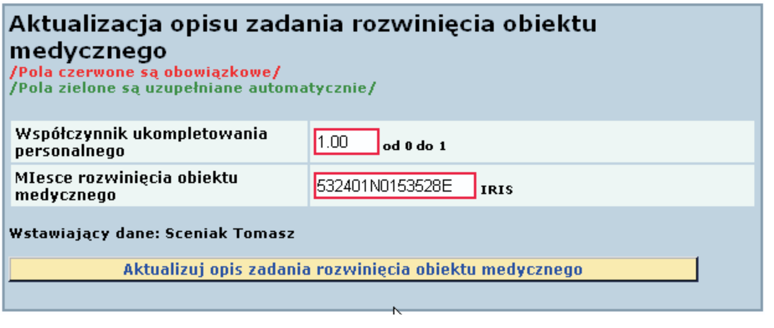


Fig. 4. Example of support units interface

It is possible to identify losses on battlefield in real time and show results on charts (Fig. 5) and to show engineers activity on battlefield map (Fig. 6).

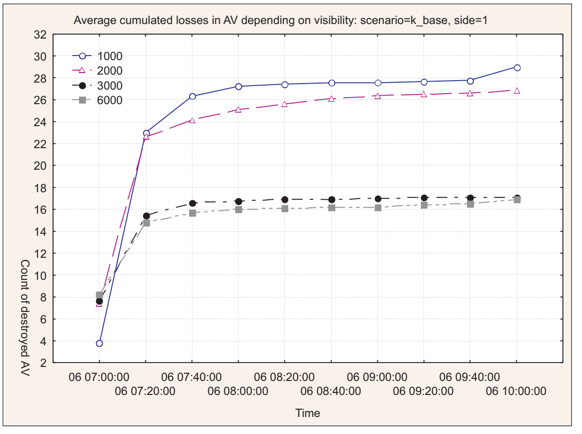


Fig. 5. Average cumulated losses in armoured vehicles (AV) depending on visibility range at day

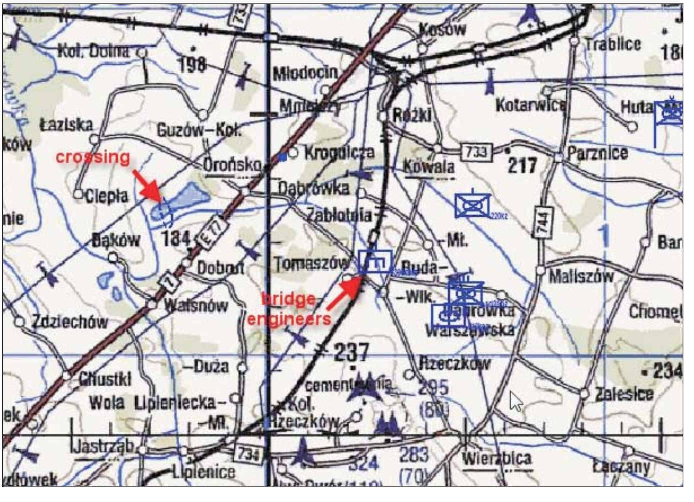


Fig. 6. Tactical situation of bridge engineers units

We can describe many parameters of engineering units using units interface (Fig. 7).

A structure of support units can be in any moment shown for decision makers (Fig. 8).

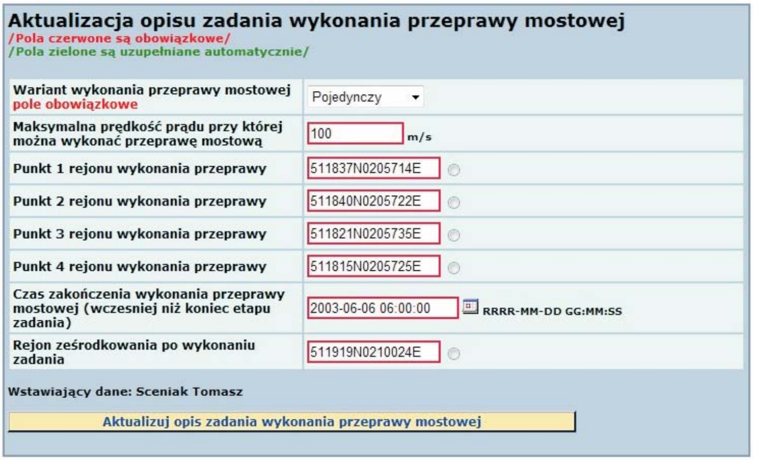


Fig. 7. Example of bridge engineer units interface

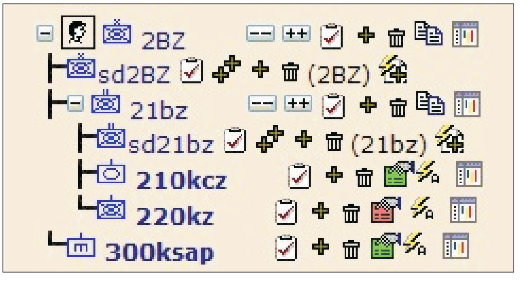


Fig. 8. Example of unit structure

8. Conclusions

In the article, a systematic approach to the model and design support activity on battlefield was presented. The described method was worked out and applied to analyze, design and implement engineering, technical and medical support during development of simulation software for Simulation and War Games Centre of Polish Armed Forces.

Received April 10 2008, revised November 2008.

REFERENCES

[1] A. Najgebauer, T. Nowicki, *The methodology of modelling and interactive simulation of combat processes for CAX and DSS*, Proceedings of SPIE — The International Conference on Enabling Technology for Simulation Science IV — AeroSense 2000, 25-27 April 2000, Orlando, USA.

[2] T. Nowicki, A. Najgebauer, *Modelling and analysis of conflict situation in a distributed interactive simulation environment*, Proceedings of System-Science Conference, Wrocław, Poland, 15-18 September 1998.

[3] T. Nowicki, A. Najgebauer, *The local combat model in a simulation environment*, Proceedings of System-Science Conference, Wrocław, Poland, 15-18 September 1998.

T. GÓRSKI, T. NOWICKI, R. WANTOCH-REKOWSKI

Modelowanie i implementacja wsparcia działań wojsk lądowych dla zdarzeniowej symulacji walki

Streszczenie. W pracy przedstawiono zagadnienie modelowania działań jednostek wsparcia wojsk lądowych. Nacisk położony został na podstawowe działania w zakresie wspomagania inżynieryjnego, obsługi technicznej i medycznej, na przykład rozbudowy fortyfikacyjnej, budowy zapór inżynieryjnych, procesów technicznego obsługiwania, poszukiwania rannych i zabitych, itd. Zaprezentowana została metoda modelowania i projektowania tego typu działań. Pokazano modele matematyczne działań jednostek wsparcia inżynieryjnych, obsługi technicznej i medycznych oraz modele zadań realizowanych przez te jednostki. Zunifikowano zarówno opis jednostek inżynieryjnych, obsługi technicznej i medycznych, jak i opis realizacji zadań wykonywanych przez te jednostki. Unifikacja ta, przedstawiona w pracy, jest interesująca z punktu widzenia różnorodności tego typu jednostek.

Słowa kluczowe: model symulacyjny, modelowanie działań wojsk lądowych, jednostki wsparcia
Symbole UKD: 355.077.1