Abstract

The paper deals with some aspects of designing and using deterministic and stochastic simulators for military trainings. These aspects are divided into three groups: (1) connected with experiences of authors concerning the usage of deterministic and stochastic models in simulators for military trainings; (2) connected with designing simulators for military trainings using simulation design patterns, (3) connected with a set of tools, which enables fast preparation of a game scenario, calibration of simulation models and evaluation of simulation results.

1. Introduction

During designing military simulators (but not only for such software) the main problem is to identify specific problems and define methods of solving them from the context point of view [6]. The main context is the application of the simulator that is being designed. The context also includes a software model abstraction level, which the problem applies to. Therefore, we can distinguish the levels of software basing on the levels defined by the Model Driven Architecture (MDA) [9], [11] and use some software design patterns [8].

Building simulation models for combat simulators requires constructing combat models: deterministic or stochastic. The main idea of constructing stochastic combat models lies in computer simulation of complex processes that are realized in the battlefield. The stochastic character of such processes results from many factors, among them [7], [12]:

- a large number of known and unknown factors that affect the course of the process,
- uncertainty and incompleteness of information about the nature of the process,
random influences of terrain and atmospheric factors,
- influences of the human factor including, but not limited to morale, training and decisions made by leaders,
- random character of the firing outcomes.

In many of the existing simulators (e.g. CBS [4], JTLS [16], ModSAF [3], [5], OneSAF [14], Zlocien [2], [13], [15]) different types of combat models are used. The model of the combat attrition process is most often based on the well-known Lanchester model. Modern simulators, which are used for Computer Assisted Exercises (CAX) should have a set of tools, which enables fast preparation of a game scenario, calibration of simulation models and evaluation of simulation results. One of the CAX steps is the scenario preparation. During scenario preparation it is necessary to prepare many kinds of data: military equipment, weapon and ammunition parameters; database of terrain; military unit parameters, its position and initial task. Before the main exercise usually there are many tests called miniCAX. In the paper, such set of tools for exemplified simulator is presented. Moreover, simulation tools described in the paper can be used for:
- optimization of military units command chains,
- evaluation of military operational rules and improving command and control procedures,
- research of military equipment parameters changing impact on results of military actions,
- quality of battlefield processes models verification (shooting, target searching, movement, etc.)

The paper is organized as follows. In the first section we describe two exemplified simulators: stochastic (SBOTSS Zlocien) and deterministic (DCombSim) which have been built at Cybernetics Faculty of Military University of Technology in Warsaw (Poland) and authors of this paper are members of the team which has done them. SBOTSS Zlocien has been put into practice at War Games and Simulation Center of National Defence University in Warsaw. This system has been used during CAX exercises. DCombSim is a part of some Decision Support System (DSS) which supports Polish C4ISR systems.

### 2. Description of stochastic and deterministic simulators and their combat models

In this section we present two simulators: stochastic (SBOTSS Zlocien) and deterministic (DCombSim) which have been built at Cybernetics Faculty of Military University of Technology in Warsaw (Poland) and authors of this paper are members of the team which has done them. SBOTSS Zlocien has been put into practice at War Games and Simulation Center of National Defence University in Warsaw. This system has been used during CAX exercises. DCombSim is a part of some Decision Support System (DSS) which supports Polish C4ISR systems.

**Stochastic simulator description**

The stochastic simulator being considered is the Simulation Based Operational Training Support System (SBOTSS) – Zlocien. The detail description of the Zlocien is presented in Table 1.

Zlocien is an integrated, interactive, multi-sided land, analysis and training support model (with logistics, engineering, electronic warfare and intelligence functions), which realizes stochastic ground-combat attrition. The system is a federation, High Level Architecture (HLA) compliant [10], and cooperates with C3 system, heterogeneous platform (Sun Solaris, Windows NT).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Land operations, corps-division-brigade levels. Supported by detailed logistics and integrated intelligence operations, air support, EW.</td>
</tr>
<tr>
<td>Span</td>
<td>ADRG digitized maps and VPF terrain data permit the model to be used worldwide. The Terrain Rectangle Model (TRM) and Railway_and_Road Net Model (R&amp;RNMM) can be used to build terrain files to support the Zlocien model.</td>
</tr>
<tr>
<td>Environment</td>
<td>Rectangle-based terrain aggregates regional terrain and environmental characteristics: traffic-ability, elevation, vegetation, chemical contamination, and weather – granularity is 200mx200m. Railways and roads are mapped via</td>
</tr>
</tbody>
</table>
independent railway_and_Road Net Model, which is complementary to Terrain Rectangle Model. Specific terrain or engineering objects are modeled separately and can be located on the maps transformed by terrain models – TRM and R&RNM.

Software

Combat simulator, After Action Review (AAR) procedures, Calibrator, Set of DBs (operational, terrain, scenario), Scenario Editor, Portal SBOTSS Zlocien, Reporter AAR, Visualization Server, ADatP3 Editor.

Deterministic simulator description

The deterministic and discrete time-driven simulator DCombSim models two-face land conflict of military units on the company/battalion level. The simulator is implemented in JAVA language as an integrated part of some systems for CAXes. The model concerns a couple of processes of firing interaction and movement executed by a single military unit. These two complementary models use a terrain model described by a network of square areas, which aggregates movement characteristics with 200m×200m granularity. The course of each process depends on many factors among them: terrain and weather conditions, states and parameters of weapons the units are equipped with, type of executed unit activities (attack, defense), distance between opposing units.

Presented in Fig.1 is the visualization of an exemplified redeployment variant in DCombSim.

Stochastic and deterministic combat models

The stochastic combat model consists of two main and complementary processes: target searching and firing. The processes describe the interaction between military units on the company level. The military unit has groups of fighting homogeneous elements, each of them carry out searching and firing processes under optical (electromagnetic) visibility. The single fighting element may be in one of the three stages: idle, searching or firing. The times of the transition between different stages are random variables.

The mathematical model of the targets detection activity is the random variable, which describes the amount of time to single target detection under fixed conditions:

\[ P \{ T_{td} < t \} = 1 - e^{-tdi(i,j)\cdot t} \tag{1} \]

where:

- \( tdi(i,j) \) - mean value of target detection intensity, where observers (single weapon personnel) of the i-th unit are detecting targets (single weapon) of the j-th unit,
- \( tdi_{nom_{time_of_day}} \) - nominal target detection intensity is a function of time of day (day, night),
- \( N_o(i) \cdot N_t(j) \) - number of observers of the i-th unit, number of targets of the j-th unit,
- \( f_{fort}(j) \) - degree of fortification of the j-th unit,
- \( f_{on_foot}(j) \) - degree of forces dismounted of the j-th unit,
- \( dist(i,j) \) - distance between the i-th and the j-th units,
- \( DR_{max}(i) \) - maximal detection range of the i-th unit,
- \( f_{vel}(v(i),v(j)) \cdot f_{vel}(v_o,v_t) \) - coefficient on detection influence of observers and targets velocity,
- \( f_{top}(i,j) \) - degree of visibility is a function of terrain (topographical) conditions between the i-th and the j-th units.

The deterministic land combat model describes two complementary processes of firing and movement on the company level. The combat attrition process is based on the well-known Lanchester model. The strength of the opposing forces is described by the continuous variables such as...
as combat power vector dedicated for each weapon system type e.g. armored vehicles, tanks, artillery, antiaircraft system, infantry. The combat power of the conflict side for the particular armament type is calculated based on its parameters and weapon system quantity the side is equipped with. The amount of the combat power losses are evaluated separately for each armament type in every single simulation step. The course of the simulation processes depends on the human interaction which consists of appointing tasks. The task is described by the route for a unit and their activity type e.g. attack, defense, march.

3. Designing simulators for military trainings using simulation design patterns

The second group of aspects deals with designing simulators for military trainings using simulation design patterns.

The essential question during simulation systems designing is identification of specific problems for simulation systems and defining methods of solving them from the context point of view. The main context is the application of the simulator being designed. The context also includes a software model abstraction level, which the problem applies to. We can distinguish the levels of software basing on the levels defined by the Model Driven Architecture (MDA) [8], [11]: Computational Independent Model (CIM), Platforms Independent Model (PIM), Platforms Specific Model (PSM).

Model of the CIM level is relatively difficult to codification, therefore it will be skipped. The model of the PIM level consists of two parts: the analytic model, which defines the responsibility range of the individual software blocks, and the design model, which is focused on the qualification that project constructions have to carry out the individual responsibilities of the analytic model. The profile of the typical requirements is the design patterns context of this simulation system model level. The model of the PSM level includes project elements from the PIM level integrated with the specific implementation environment. This environment consists of such elements as follows: framework solutions, libraries, commercial components, used middleware solutions, etc. Additionally the implementation model takes into account requirements of the specific software production environment, for the specific programming language implementation. The profile of the implementation and execution environment is the design patterns context of the PSM model level.

For discrete event simulators it is possible to set typical problems essential for analyst and simulator architect. These problems can be divided into three groups: (1) patterns of objects service and their resources; (2) patterns of processes service, which realizes the behaviour of simulation objects; (3) patterns of events and interactions. The first group of patterns consists of the following elements: invoke simulation object, delete simulation object, declare behaviour, declare resource, do action, update resource, declare resource monitor, other. The second group of patterns consists of the following elements: run process, terminate own process, terminate other process, terminate subprocess, send signal, wait for signal, wait duration, wait for subprocesses termination. The third group of patterns consists of the following elements: create event, declare event service, serve event, create interaction, send interaction, declare interaction service, serve interaction, other.

4. Set of tools in SBOTSS Zlocien for preparation and evaluation of a game scenario

In this section described are a set of tools, which enable fast preparation of a game scenario, calibration of simulation models and evaluation of simulation results.

Scenario Editor is one of the Computer Assisted Exercise elements. Scenario Editor allows preparing a scenario, which consists of:

- unit names,
- unit position,
- unit initial tasks,
- unit equipment and weapons,
- chain of command.

There are some steps to prepare a scenario. First of all it is necessary to prepare a database of basic units and other parameters.

The scenario editor allows preparing the description of the main kinds of weapons (see Fig. 2). Each weapon is described by: strength (potential), max speed, typical speed on ground, typical speed on water, caliber, shooting parameters, tracked or wheeled vehicle, kind of fuel, crew, other parameters.

The next step is to prepare templates of all kinds of units (e.g. armored, motorized, support, technical support, medical support, air forces, engineer and so on, see Fig.3). Each unit is described by: name, kind, equipment, weapons, number of soldiers, march column length, level, average speed, other.
Command posts are described similar to unit templates (see Fig. 4).

One of the main functions is the “Structure templates”. This function allows to prepare different kinds of command chains templates. Fig. 5 shows an example of two units: armoured unit “bcz” consists of PT91 tanks (four armoured companies, command post, supply platoon and maintenance platoon) and reconnaissance unit “krozp” (command post and three reconnaissance platoons). Fig. 6 shows example of opposite forces.

Each weapon has its shooting parameters (see Fig. 2). Because the simulator uses stochastic combat models it is necessary to fix kill probability.

Fig. 7 shows a small part of the all kill probability chart. This chart consists of kill probability for all kinds of weapon and used ammunition defined earlier. Kill probability is defined only for this kind of weapon, which can “see” targets in its range of fire (e.g. guns, tanks). For other kinds of weapon like artillery, howitzer or mortars it is necessary to define the strike range (see Fig. 8).
Each vehicle has its specific loading parameters. Figure 9 shows, which ammunition can be loaded on defined in the system vehicles.

The main function of the system is scenario preparation. Figure 10 shows the main window of the Scenario Editor. The scenario consists of: military units (equipment, weapons, position), initial tasks, command chain, other.

Fig. 11 shows an example of the opposite forces command chain.

During scenario preparation the user can also use an application called “Mapnik”. This application allows fixing positions of all military units (command chain) and visualizes it using different kinds maps (see Fig. 12 and 13).

The last step of the scenario preparation is the initial task preparation. Scenario Editor allows prepare tasks as follows: attack, march, defence, supply, combat reconnaissance, bombardment, technical support, medical support, electronic
warfare, air attack, chemical attack, chemical reconnaissance, military engineering and other.

Fig. 14 shows elements of march: departure area, target area, initial marching point, control march points.

The After Action Review Analyzer is a tool which enables checking selected game characteristic values off-line, after simulation. AAR Analyzer is a Microsoft Excel 2003 application based on pivot tables and Visual Basic for Applications procedures.

The characteristics are divided into a few categories as follows: state/resource levels of units (weapons, personnel, materials, potential), losses, distribution of units regarding weapons, personnel, materials, potential (time distribution, structural distribution, time-structural distribution), schedule for units recognized by other units, task realization degree, real action speed, relation between real action speed and normative speed. AAR Analyzer allows the presentation of simulation results using tables and graphs.

The main window of the AAR Analyzer is shown in Fig. 15. The first group of characteristics is state/resource levels of units (weapons, personnel, materials, potential). In this group we can set the following parameters to obtain selected state/resource levels of units: scenario name, conflict site name, type of weapon/material, units names, number of experiments run for scenario, time window.

Presented in Fig.16 and Fig. 17 is the state (level) of selected weapons for selected units (table and graph). Presented in Fig.18 and Fig. 19 is the state (level) of selected petrol for selected units (table and graph). The second group of characteristics is losses distribution of units regarding weapons, personnel, materials, potential (time distribution, structural distribution, time-structural distribution). The time distribution of losses allows us to analyse losses in time. The structural distribution of losses allows us to analyse losses for selected type(s) of weapon/material. The time-structural distribution of losses allows us to analyse losses for selected type(s) of weapon/material in selected time period. Presented in Fig.20 and Fig. 21 are time-structural losses of tanks for the “RED” conflict site.
The next characteristic – relation between real and normative action speed allow us to calculate real action speed/normative action speed ratio. Real action speed is calculated by taking into account average speed of units movement during action. Normative action speed is taken from normative tables, which take into account: kind of terrain, relation between potential of opposing forces and type of defence (temporarily prepared, prepared in advance). In Fig.22 we have real action speed equaling 0.93 km/h, normative action speed (for scenario conditions) 0.15km/h and calculated speed ratio – 6.23.

Task realization degree is a characteristic, which allows us to calculate values of some conditions and compare them with acceptable values. We set acceptable values of percentage losses of units potential and personnel (own or opponent) and then real values (based on scenario simulation results) are calculated. For each condition we obtain a mark: “satisfied” (green) or “not satisfied” (red). In Fig.23 presented is the calculation of the task realization degree.

The AAR Reporter is a tool, which enables checking selected game characteristic values (subset of characteristics from the AAR Analyzer) on-line, during gaming as well as after simulation.
The calibrator in SBOTSS Zlocien is a tool, which enables (before running simulation scenario) fixing values of calibration parameters that have influence on many of simulation models (see Fig.24).

It includes 126 parameters, which are grouped in 8 categories and their values are written (read) in (from) calibration database. For each of these parameters the default value has been proposed (after pre-calibration), but it is possible to change values of these parameters separately for each of the simulation scenarios. The most important categories of the calibration parameters are as follows:

- parameters of detection category (16 parameters) – coefficients, which have influence on intensity of target detections depending on: time of the day (night), velocity of target, velocity of observer, terrain roofing (buildings, forests) and relief;
- slowing down rate of actions category (14 parameters) – coefficients, which describe the slowing down rate of actions depending on the kind of vehicle, kind of terrain soil and time range of precipitation;
- rate of battalion advance category (43 parameters) – coefficients, which describe the rate of battalion advancement depending on: kind of the terrain, degree of engineering preparation of defence, strength of opponent (military potential);
- action capabilities category (18 parameters) – these coefficients describe thresholds of units military potential - below these values units lose its action capabilities or they are broken.

The calibration tool is more precisely described in [1].

5. Summary

In the paper, it was written and discussed that, for discrete event simulators, it is possible to set typical problems (as software patterns) essential for an analyst and the simulator architect. These problems have been divided into three groups: (1) patterns of objects service and their resources; (2) patterns of processes service, which realize the behaviour of the simulation objects; (3) patterns of events and interactions. Simulators, which are used for CAX’es should have a set of tools, which enable fast preparation of a game scenario, calibration of simulation models and evaluation of simulation results. A set of tools, which enable fast preparation of a game scenario, calibration of simulation models and evaluation of simulation results have been discussed. For example, the Scenario Editor which has been presented is a tool that extremely decreases the preparation time of CAX. Moreover, calibration of simulation models of complex processes is a very important process, which enables to tune these models. It is very responsible work and should be done very carefully using specific tools. In the paper, the calibration tool in SBOTSS Zlocien has been briefly described (it is precisely described in [1]).

A new version of the SBOTSS Zlocien is preparing for MS Windows system using PC platform. Currently works to inter-connect SBOTSS Zlocien, Joint Theater Level Simulation [16] and ABS2 [17] using HLA interface are carried out.

6. Acknowledgements

This work was partially supported by research project numbers PB-MNiSW-0055/R/T00/2008/06 titled (in Polish): "Symulacja działań bojowych w heterogenicznym środowisku wielorozdzielczej i rozproszonej symulacji" (Combat simulation using heterogeneous multiresolution and distributed environment) and PB-MNiSW-0050/R/T00/2008/06 titled (in Polish): “Integracja systemów dowodzenia” (Integration of command and control systems).
7. References


