Methods and tools for generating high detail terrain model utilized in movement and behavioral models, reflecting border control actions.

MARIUSZ CHMIELEWSKI, PIOTR STĄPOR, JAKUB KĘDZIOR, MARCIN KUKIEŁKA
Cybernetics Faculty
Military University of Technology
gen. S. Kaliski Street 2, 00-908 Warsaw
POLAND
mchmielewski@wat.edu.pl, pstapor@wat.edu.pl , jkedzior@wat.edu.pl, mkukielka@wat.edu.pl

Abstract: This work summarizes a novel approach for discrete terrain model construction and algorithms utilizing the terrain and road network data. The models and algorithms are baseline of SymSG Border Tactics constructive simulator focusing on border control operations performed by Border Guard forces involving local population, perpetrators and other actors. Developed simulation algorithms include simulation objects movement, reconnaissance process, patrolling, pursuit, assault, identity checks, etc. and require discrete terrain model supplemented with specific characteristics delivered as coverages. SymSG Border Tactics is a distributed simulation software dedicated for tactical training including high-resolution law enforcement and opponent modelling as well as their interaction design. Required level of details within the models (single officer) enforced high detail terrain and behavioral models. Large scale of GIS data required for the simulator to perform adequate exercise execution has been a challenge. The novelty of the presented solution is hybrid approach which involves a discrete square-based terrain supplemented by the multi-layer network model for road and track system. Due to required resolution of terrain models (cells 10x10m) for larger scenarios up to 50x20km an efficient terrain representation model had to be designed followed by multi-criteria path finding algorithms. This paper presents and summarizes the process, methodology and tools which use open GIS data sources (OpenStreetMap.org) to generate terrain model further utilized by hybrid multiresolution movement algorithms.

Key-Words: high detail model, constructive simulation, border guard training, open street map, multi-resolution path finding, hybrid algorithms

1 Introduction and scope

The constructive simulation is a tool broadly utilized in case of operational command staff training in military but also gains new areas of application in border protection activities. Reflecting activities connected with command or control procedures require tools which concentrate on implementing decisions in form of tasks presented to interacting simulation objects such as units, groups and individuals representing forces, authorities, social groups and perpetrators (criminals, terrorists, etc). Simulation games can be used to support decision-makers' training processes and to evaluate developed courses of action for subordinate forces and resources. The modeling of high-intensity armed conflicts delivers the use of aggregated simulation models, but technology evolution forced, a higher resolution and accuracy of the modeling. The SymSG Border Tactics project focuses on the development of a multi-resolution simulation environment designed to train Border Guard commanders and officers in planning and managing border control operations. One of the major group of requirments for constructive simulator is connected with the resolution of simulation models and required data driving objects' state calculations. Border Guard operations are performed on tactical level involving often individual officers and perpetrators thus requiring high detail terrain, movement algorithms and interactions. Adequate simulation algorithms for forces movement, reconnaissance, patrolling, pursuit, assault, identity checks, etc. require discrete terrain model supplemented with specific characteristics (water, forest, road, urban area, etc.) coverages. This paper presents and summarizes the process, methodology and tools which use open GIS data sources to generate terrain model further utilized by hybrid multiresolution movement algorithms. Presented method implements requirements formulated for SymSG constructive simulator, which have been briefly described in the paper explaining the system's vision and construction aims.

2 OSM role in terrain extraction

The terrain data used in SymSG Border Tactics simulator originates from an Open Street Map (OSM) database. It has been extracted and then specially processed with the help of a dedicated tool – The SymSG Terrain Extractor.

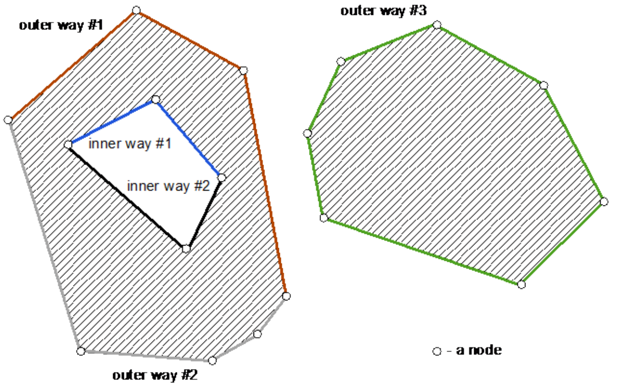
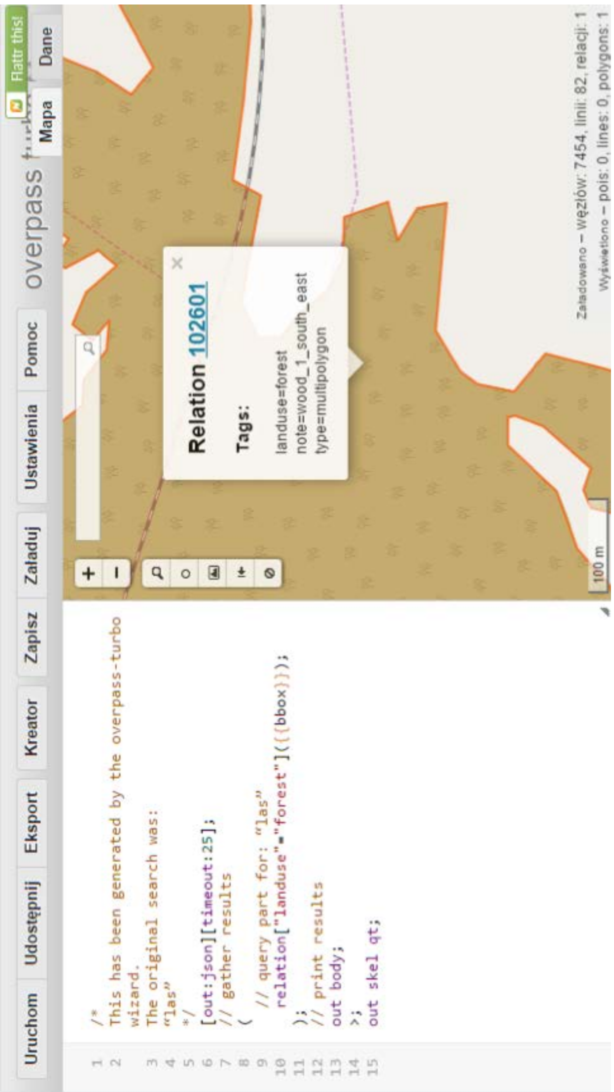


Fig.1. An example of a complex relation structure.

OSM itself provides a free access to geospatial data using query languages via webservices. The structure utilized there is as follows: the most basic object on the map is a point; the points gathered into a sequence (polylines) are called ways; finally, relations of type multipolygon are complex objects formed from other relations and waypoints. To be more specific, such relation consists of two sets of child elements, to which objects are assigned according to their role attribute: “outer” or “inner”. The first set encompasses shapes’ perimeters divided into fragments, while the latter consists of chunks of internal boundaries i.e. inner shapes representing empty spaces. Neither “outer” nor “inner” sets are ordered, and what is more, we also do not get information about how many separated figures or hollow spaces are there. In most cases the elements from both sets had to be assembled by an algorithm, which tried to join polylines basing on their endpoints.

Each object in OSM is discriminated by following two features: type and a numerical identifier. Its description is contained in a form of tags list consisting of key-value pairs. E.g. in order to get woods one must search for relations with key ”landuse” and corresponding value “forest”. There is also one shortcoming in query processing itself - that is, if at least one fragment of an object is not included in the area marked for search, it will not be returned in response by Overpass API. For example, the lake below will not be returned in a query for water layer.



Relation 102601

Tags:

landuse=forest
note=wood_1_south_east
type=multipolygon

Załadowano – węzłów: 7454, linii: 82, relacji: 1
Wyświetlono – pois: 0, lines: 0, polygons: 1

```
/*
This has been generated by the overpass-turbo
wizard.
The original search was:
“las”
*/
[out:json][timeout:25];
// gather results
(
  // query part for: “las”
  relation["landuse"="forest"]({{bbox}});
);
// print results
out body;
>;
out skel qt;
```

Fig.2. Running an exemplary query in online overpass API based porta – overpass-turbo.eu.

For the purpose of addition and retrieval of geospatial data, a webservice has been provided by OSM creators. Its usage, however, was problematic due to some of its limitations. As authors of OSM API noted: it was not intended for downloading large quantities of data [18]. Finally, Terrain Extractor has been designed to make use of Overpass Turbo webservice, which takes advantage of Overpass API, an alternative programming interface optimized for data consuming (instead of editing), with a powerful built-in query language. It encompasses features of previously available OSM API extension – XAPI. [19]

It is worth to note some intrinsic problems of OSM as well as its accompanying webservices. From time to time the webservice refuses to respond to extensive queries. Moreover, there exists a limit of data for a single query that can be retrieved (even in Overpass API. An impactful obstacle in data processing automation lays in poor structuralization of OSM data, which has two sources: 1) ambiguous and mediocre data base design, which seems to be stiff as it has to preserve legacy old records; this includes storage of multiple values in one attribute often without explicit formal grammar; 2) database is periodically updated by volunteers without a necessary supervision; as a result OSM database contains some multi-polygons that do not comply with formal specification and thus raise various errors that cause processing problems.

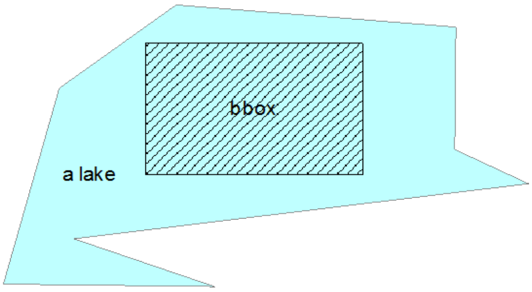


Fig.3. One of problems with overpass API – for a given bbox (rectangular query area) no water layer objects will be returned.

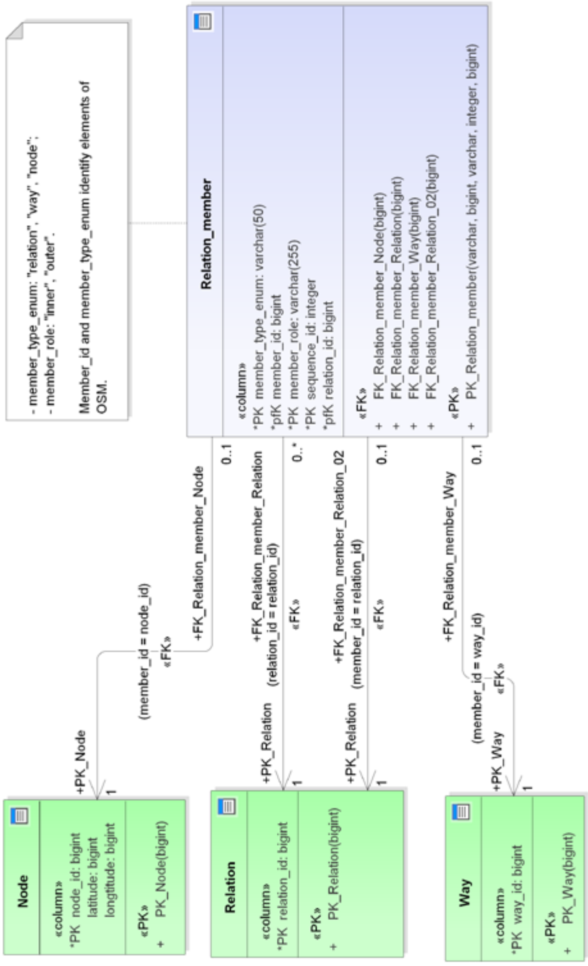


Fig.4. Diagram of elements in OSM database that are important for data extraction (reverse engineered schema) – Part 1.

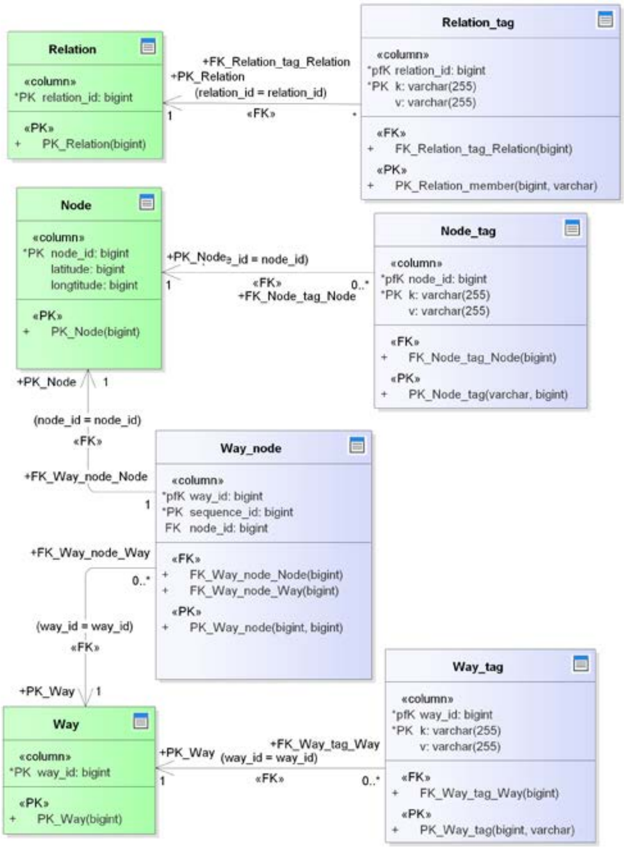


Fig.5. Diagram of elements in OSM database that are important for data extraction – Part 2.

On the other hand, relation validating tools exist and the OSM community tries steadily to correct the errors, albeit it takes significant time. Most of these problems cannot be resolved by automatic process, thus Terrain Extractor offers a user configurable list of objects (type and id) that are to be omitted during data processing.

Obtaining the area covered with sea water within OSM is needlessly complicated, due to a specific way it represents oceans and seas in its database. For example, acquiring Baltic Sea's shore line for Poland requires to process coastline polylines' fragments of the entire Word Ocean. The terrain found on the right-hand side of a given coastline section, while traversing its nodes in a specific order is to be considered water.

3 Terrain acquisition process

When the extraction process commences, the OSM Terrain Extractor generates and sends an appropriate query written in Overpass XML language.

Only terrain altitude data (the height above sea level for a certain latitude and longitude coordinates) is taken from different source – DTED files as OSM does not contain such information.

Objects that ought to be included in each layer can be defined earlier via settings dialog. User can set there the type of objects to look for (relation or way) with associating key and value.

Table 1. An example of objects included into layers

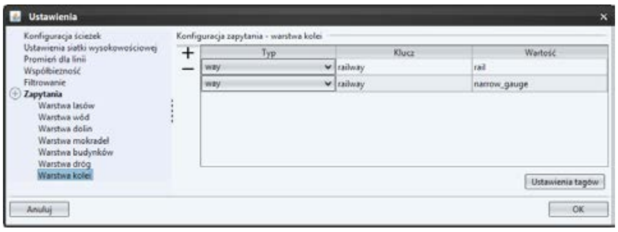
layer	key	type	value
Forest layer	landuse	way	forest
		relation	forest
Water layer	natural	way	water
		relation	water
	waterway	way	river
			riverbank
			steam
			ditch
Valley layer	natural	way	sinkhole
			valley
…	…	…	…

The query includes also a rectangular border (bbox) that defines the area to be taken into account during the search process. Below we have provided an example of such search definition for forest data layer. After a set of all criteria-fitting elements (union) is created. The “recurse” keyword is then used to fetch recursively all child elements – nodes of ways in union, ways that belong to relations in union. Note that output format is set to json.

```
<osm-script output="json" timeout="180">
  <union into="_">
<query into="_" type="way">
     <has-kv k="landuse" modv="" v="forest"/>
     <bbox-query n="52.1600" into="_" w="21.0600"
                  s="52.1500" e="21.0700"/>
   </query>
<query into="_" type="relation">
     <has-kv k="landuse" modv="" v="forest"/>
     <bbox-query n="52.1600" into="_" w="21.0600"
                  s="52.1500" e="21.0700"/>
   </query>
  </union>
  <print e="" from="_" mode="body" n="" order="id" s="" w=""/>
  <recurse from="_" into="_" type="down"/>
  <print e="" from="_" mode="body" n="" order="quadtile" s=""
w=""/>
</osm-script>
```

Fig.6. A query generated for a forest layer by Terrain Extractor.

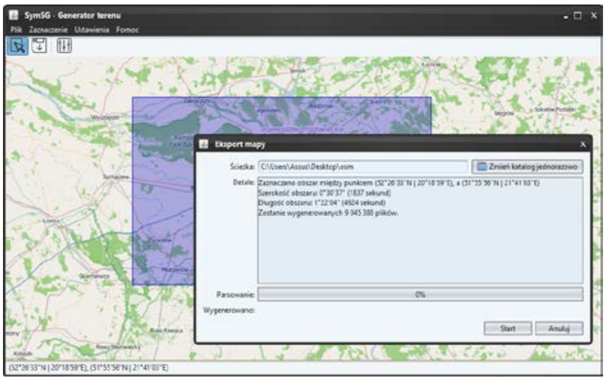
After fetching the data from the web service the response json text is converted with the help of gson into RelationOSM, NodeOSM, WayOSM object classes. All these objects then go to a MapData instance (one instance per each layer) that stores them in hashtables.



Konfiguracja zapytania - warstwa kolei

Typ	Klucz	Wartość
way	railway	rail
way	railway	narrow_gauge

Fig.7. A Terrain Extractor settings dialog allows user to define which geospatial objects belong to a given layer.



Ścieżka: C:\Users\Assus\Desktop\esm

Detale: Zaznaczano obszar między punktem (52°26'33"N | 20°18'59"E), a (51°55'56"N | 21°41'03"E)
Szerokość obszaru: 0°30'37" (1837 sekund)
Długość obszaru: 1°22'04" (4924 sekund)
Zostanie wygenerowanych 9 045 388 plików.

Parsowanie: 0%

Wygenerowano:

(52°26'33"N | 20°18'59"E), (51°55'56"N | 21°41'03"E)

Fig.8. A main Terrain Extractor window with a selected processing area and export data dialog.

For further processing, Terrain Extractor employs jts library, which allows performing various geometric operations and produces String type output that can be visualized in JTS Test Builder tool.

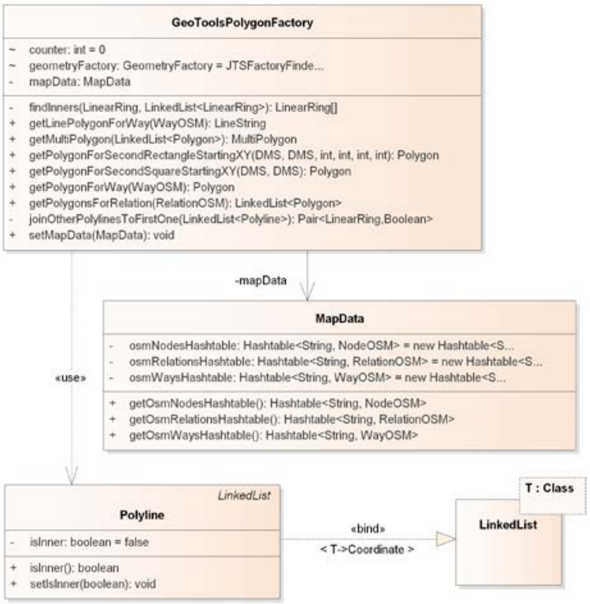


Fig.9. Representation of OSM objects in memory in Terrain Extractor before conversion to JTS library instances – Part 1.

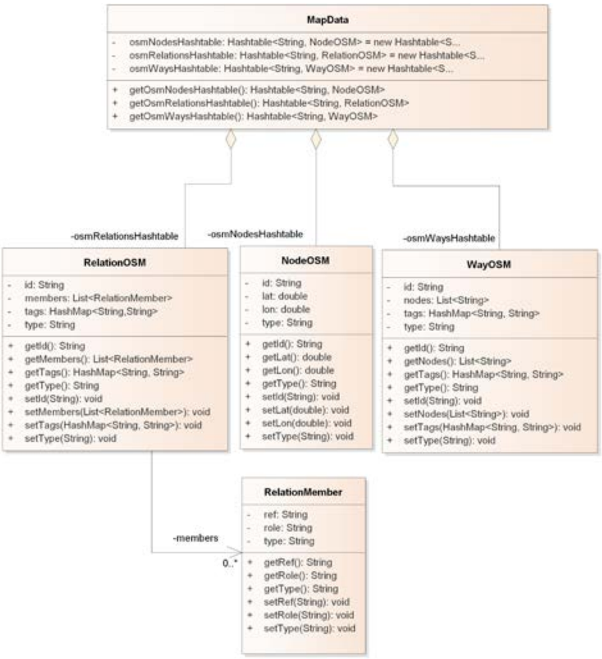


Fig.10. Representation of OSM objects in memory in Terrain Extractor before conversion to JTS library instances – Part 2.

GeoToolsPolygonFactory is a class implemented in Terrain Editor that provides means for conversion of RelationOSM, NodeOSM, WayOSM to jts library elements. The mapping goes as follows:

- nodes are represented as Coordinate instances,
- ways are translated LineString,
- relations became instances of either Polygon (polygon with possible empty "islands" inside) or MultiPolygon (geometric shape consisting of many Polygons) classes.

Creation of a Polygon object requires an instance of LinearRing (a sequence of coordinates that forms a closed loop – polygon perimeter).

Small rivers and roads, that are only represented by polylines, are then converted to 2D areas (polygons), by claiming the space in the range of a certain user pre-set radius.

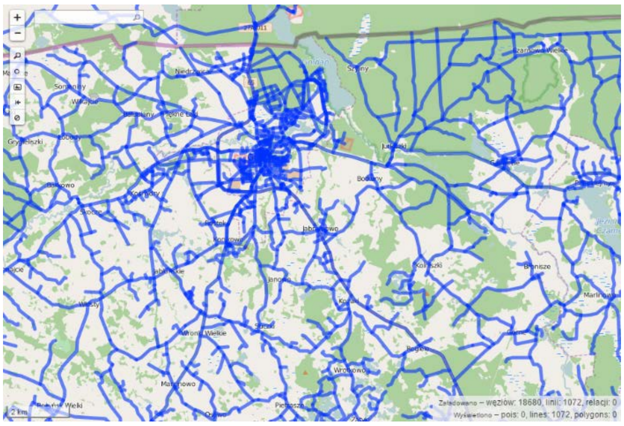


Fig.11. Road network calculated by infrastructure query displayed in developed tool utilising the-OSM API (overpass API).

Finally, each layer is represented by a union of all its shapes. Such layered “cake” map is then divided into squares 60×60 geographical seconds. To speed up the data extraction concurrent processing is applied.

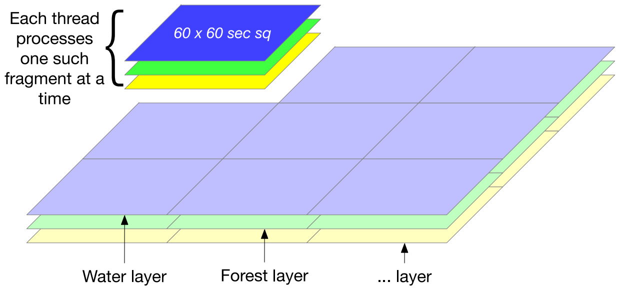


Fig.12. A layered “cake” map division. A user defined number of threads start each independently working on one “slice” square-fragment at a time.

A user defined number of threads start each independently working on one “slice” square-fragment at a time.

During each “slice” processing, the algorithm moves a square 1 by 1 geographical second sq. and for each of its position for each layer evaluates the coverage of such segment as shown in figure below.

Fig.13. Evaluating coverage for a forest layer in a slice. (Union area of square 1 by 1 second and shape in a layer is divided by area of a square.)

Currently, information extracted by the presented software for every geographical second includes:

- Buildings area coverage (per mille)
- Forests area coverage (per mille)
- Swamps area coverage (per mille)
- Infrastructure development (per mille)
- Water area coverage (per mille)

- Gullies coverage (per mille)
- Average altitude above sea level (m)
- Maximal altitude difference in sector (m)
- Dominant feature's index number.

The last step before writing data files to hard disk is the aggregation of data inside such geographical slice for different projection levels. Please refer to the next chapter for more on projection levels.

The lack of adequate standardisation of data in OSM and difficulty in reconstruction of relations as graphical objects slowed down development and proved to be an issue. What is more, some relations returned by OSM caused exceptions and only after closer study were found invalid at that time. Instead of waiting for OSM community to deal with that problems, we had implemented a dedicated instructions' set for most important (large) objects and filtered out the minor ones.

4 On terrain extraction time

In order to supply SymSG Border Tactics simulator with terrain data, a dedicated software – a Terrain Extractor – has been developed. Acquiring and parsing the data was not the only problem to be dealt with. The simulation areas are enormous - e.g. 20500 km2 for North-Eastern Poland. Also, the module responsible for pathfinding performs numerous queries, thus short-time data access became a valid requirement. A need aroused to derivate a format, which would not only use up a small amount of storage space, while keeping all the data needed by simulation, but also provide a fast access for the simulator.

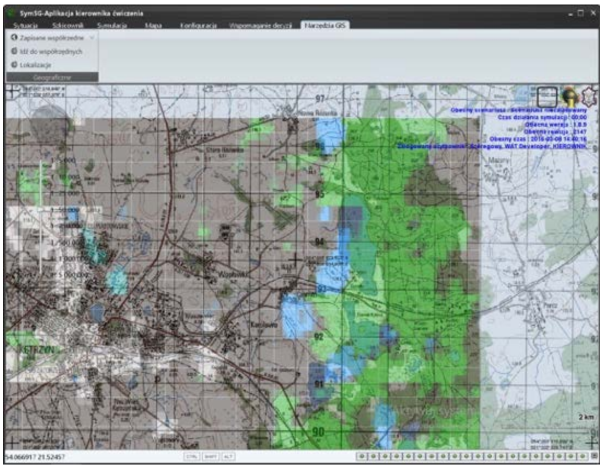
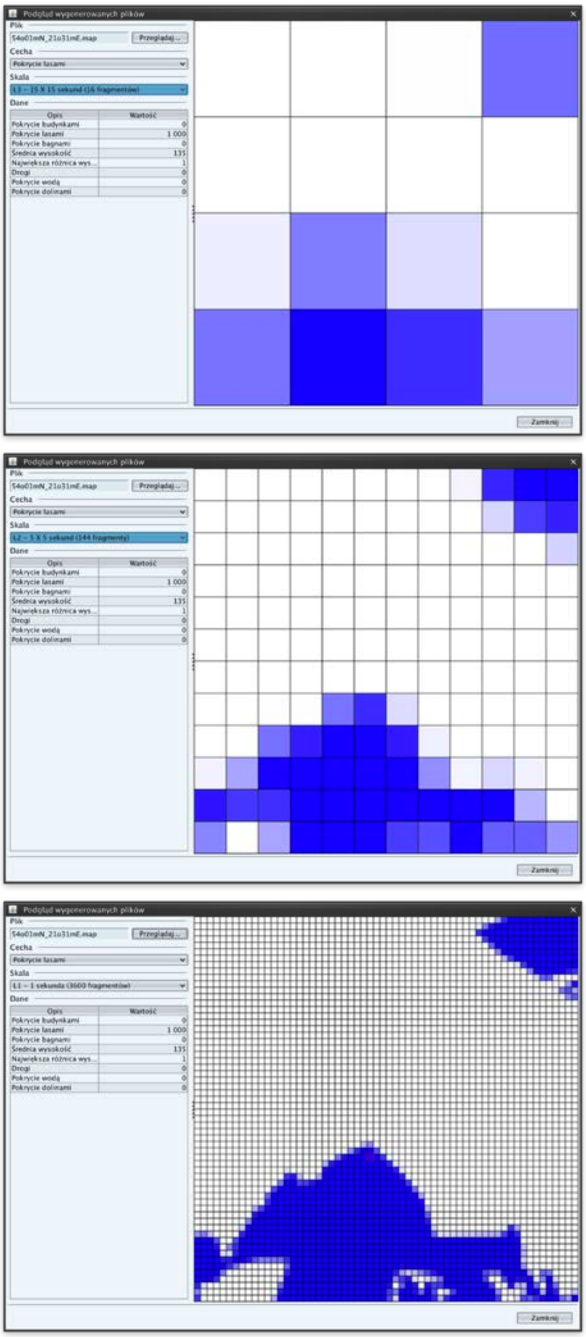


Fig.14. A fragment of extracted OSM data (simulator terrain model). Screen depicts forests' and water's layers overlyed on the CADRG topo map.



Plik
54o01mN_21o31mE.map
Cecha
Pokrycie lasami
Skala
L3 – 15 X 15 sekund (16 fragmentów)
Dane

Opis	Wartość
Pokrycie budynkami	0
Pokrycie lasami	1 000
Pokrycie bagnami	0
Średnia wysokość	135
Największa różnica wys...	1
Drogi	0
Pokrycie wodą	0
Pokrycie dolinami	0

Plik
54o01mN_21o31mE.map
Cecha
Pokrycie lasami
Skala
L2 – 5 X 5 sekund (144 fragmenty)
Dane

Opis	Wartość
Pokrycie budynkami	0
Pokrycie lasami	1 000
Pokrycie bagnami	0
Średnia wysokość	135
Największa różnica wys...	1
Drogi	0
Pokrycie wodą	0
Pokrycie dolinami	0

Plik
54o01mN_21o31mE.map
Cecha
Pokrycie lasami
Skala
L1 – 1 sekunda (3600 fragmentów)
Dane

Opis	Wartość
Pokrycie budynkami	0
Pokrycie lasami	1 000
Pokrycie bagnami	0
Średnia wysokość	135
Największa różnica wys...	1
Drogi	0
Pokrycie wodą	0
Pokrycie dolinami	0

Fig.15. Sample data and aggregation layers (15x15 seconds (top), 5x5 seconds, 1x1 second (bottom)) for forest coverage found in file 54o01mN_21o31mE.map.

Reduction of necessary data references was achieved via the introduction of multiresolution approach. While evaluating a movement path, the algorithm will utilize large-scale aggregated data first, before performing the refinement with lower resolution tiles.

Due to geoid shape of Earth $0°0'1'' \times 0°0'1''$ grid fragments resemble the shape of a trapezoid that varies with the change of latitude. One of difficulties during implementation of

SymSG software was that although terrain information is processed with accordance to geographical seconds, the simulation requires km and m to be the actual distance measuring units.

Table 2. Table of segments inside a map file

level	size in sec.	quantity	
5	60×60	1	165 aggregated segments of averages
4	30×30	4	
3	15×15	16	
4	5×5	144	
5	1×1	3600	Exact values

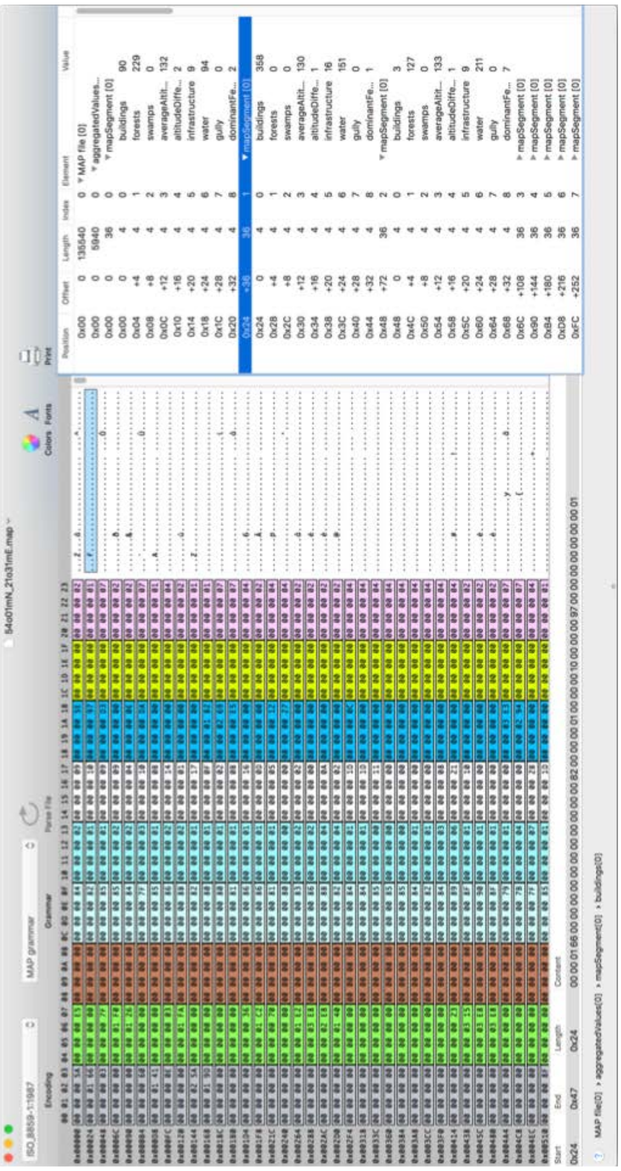


Fig.16. Browsing the insides of a map file with a hex editor.

All data is stored inside one directory. MapInfo.txt file is keeping generation dates together with border points of bbox used. The output folder also contains a set of subdirectories each corresponding to one geographical square degree. Their name corresponds to latitude and longitude in which they are found e.g. 52oN21oE.

The above described directories hold “map” files, each one encompassing data for a square with the edge of one geographical minute. One map file contains 165 aggregated data segments filled with averages for its subareas and another 3600 segments containing exact values for each second.

The aforementioned aggregations encompass data from a number of 1s × 1s squares that varies according to their scale level: 5lvl – square with a 60s-long edge, 4lvl – 30, 3lvl – 15, 2lvl -5, (lvl is a scale level assigned to 1s × 1s un-aggregated squares).

In order to minimize space usage, the files are written in binary mode. Each segment occupies 36 bytes which amounts to 136 KB per file.

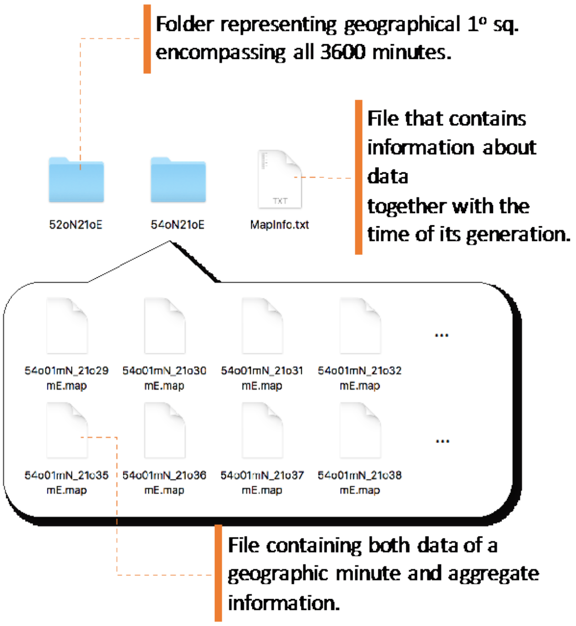


Fig.17. Output file structure of Terrain Extractor.

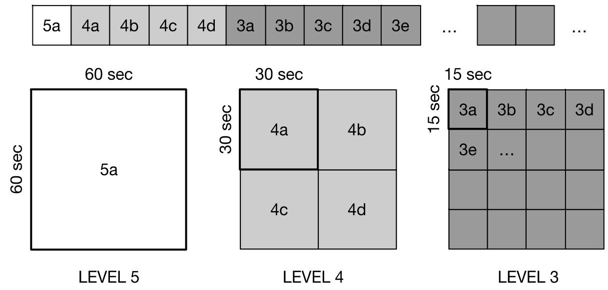


Fig.18. SymSG constructive simulator terrain model file organization.

This approach enables an efficient data access. Whenever SymSG requires terrain information for a certain coordinate, it computes a path to a map file and an offset value. After the file is accessed, only a needed chunk of data, the one starting immediately after the offset is retrieved. Such design of data together with rounding extraction to geographical minutes' grid and rectangular map fragment selection provides easy updatability as well as expansibility of terrain data. Importing another map scrap results in creation of new files and folders.

Should old and new fragments overlap, only the information at conflicting area gets overwritten while leaving the rest of previously acquired data fully usable.

4 On terrain extraction time

When measuring the extraction time, one may differentiate between the two steps: 1) polygons construction (plus obtaining the data) 2) coverage evaluation and file writing (aggregation time also goes here).

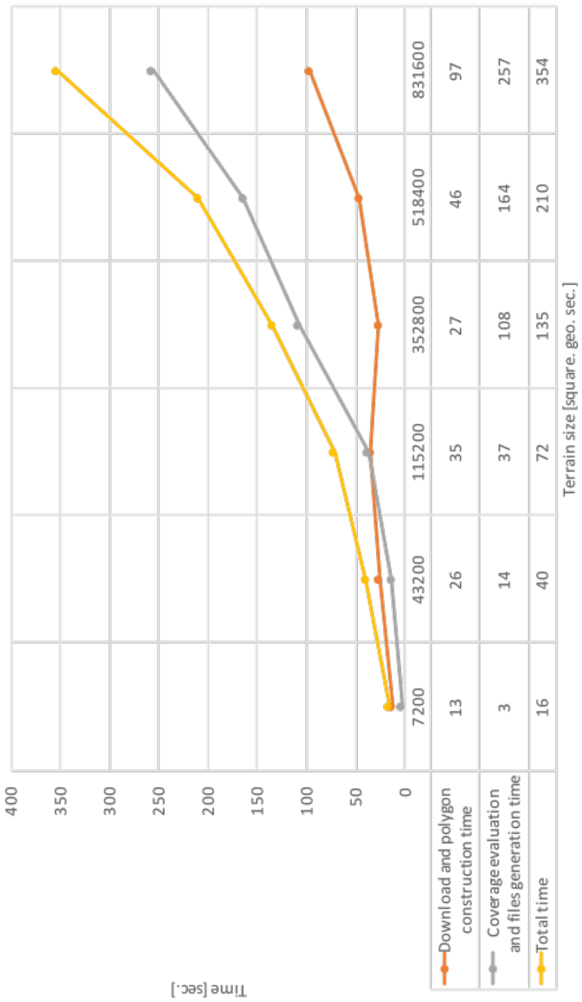


Fig.19. Sample terrain extraction times for averagely covered areas.

5 Simulation model of movement

The extraction time depends on number of factors, the most prominent ones being: 1) computer's processing power, 2) storage writing fastness, 3) speed of internet connection 4) OSM webservice occupancy, 5) size of selected area, 6) number of geographical objects inside these borders, 7) complexity of these objects.

Because of so many influences it is quite hard to predict the extraction time of a given fragment, however it must be noted that the duration of first step is quickly surpassed by the time of coverage evaluation and file writing for averagely covered areas.

One more way in which the process time would be shorten is perhaps to decrease polygons construction time by uploading child elements for each way or relation separately, thus reducing search time during the assembly.

Modelling a movement behaviour of a simulated object involves dealing with various circumstances and also choosing right priorities that are linked with specified translocation tasks. Analysis of actions carried out by armed and uniformed formations yield two distinctive classes of movement strategies: terrain and road translocation. [20]

Below, two factors have been listed that possess an impact on the movement capabilities of simulation object in a given fragment of a surface:

- maximal inclination angle between direction defined by surface and perpendicular direction to gravity vector α_{max} ,
- maximal velocity up/down a slope v_{α}.

Because objects, that are being translocated with

Table 3. Classes of movement tasks

Movement strategy	Moveable simulation object behaviour (movement task)
Terrain translocation (TTS)	Assault (kinetic actions)
	Silent Approach
	Patrolling
	Withdrawal
	Escape
	Convoy
	Short-distance relocation
Road translocation (RTS)	Medium and large - distance relocation
	Approach into troop concentration area
	Approach into operation area

RTS (mostly vehicles) use roads that must comply with legal regulations, a simplified model has been used to simulate their movement – the first factor does not play a role here, i.e. they are capable of passing impediments resulting from surface inclination. The TTS movement is achieved by a simplification – the object moves as it would be travelling through a grid made of inclined planes. The formal model that describes object moving up and down an inclined plane, is based in analysis forces distribution (i.e. thrust, friction, gravity) and some object and ramp features (i.e. mass, ramp height, ramp angle). [20]

Let us define a vehicle ω as a tuple $(\mathrm{D}, \mathrm{T}, \mathrm{G_a}, \mathrm{n}, (\mathrm{G_i})_{i=1}^{n}, \mu, \mathrm{r}, \mathrm{m}, \mathrm{C_x}, \mathrm{k}, \mathrm{w}, \mathrm{h})$ with the following properties:

- D – dynamic index (denotes driving force reserve per mass unit),
- T – maximal engine torque,
- $\mathrm{G_a}$ – axle ratio,
- n – gears number,
- $(\mathrm{G_i})_{i=1}^{n}$ – series of i-th gear ratios,
- μ – powertrain overall mechanical efficiency,
- r – dynamic wheel radius (in this case: dynamic radius of the wheel that is rolling freely without slipping),
- m – mass,
- $\mathrm{C_x}$ – drag coefficient, depends from a body shape,
- k – frontal area coefficient (frontal area as a contour projection onto a plane perpendicular to its axis),
- w – width,
- h - height.

Let surface of terrain τ cell be a tuple $(\mathrm{f}, \alpha, \rho)$ with the following properties:

- f – friction coefficient,
- α – surface inclination angle.
- ρ – air density.

Due to the fact that a considered vehicle is moving through any difficult terrain (steep slope), its speed is low, thus air resistance $\mathrm{F_p}$ and inertia $\mathrm{F_b}$ become irrelevant. The most significant forces are friction $\mathrm{F_t}$ and elevation resistance $\mathrm{F_w}$. Let $\mathrm{D_i}$ be dynamic index that denotes driving force reserve per mass unit while vehicle is moving up hill with i-th gear, then:

$$D_i = \frac{T\cdot G_i\cdot G_\alpha\cdot\mu}{r\cdot m\cdot g} \qquad (1)$$

Due to difficult terrain characteristics (steep slopes), the most beneficial case for a vehicle is availability of a maximum possible driving force reserve. It allows the vehicle to keep constant velocity and balances friction and elevation resistance. Because $\mathrm{D_i}$ is directly proportional to $\mathrm{G_i}$, maximal $\mathrm{D_i}$ values are obtained at maximal $\mathrm{G_i}$ values (while i=1). In case when a vehicle is moving up the maximal achievable surface inclined at angle $\alpha_{\max}$ with the constant speed, the thrust $\mathrm{F_N}$ balances sum of friction $\mathrm{F_t}$ and elevation resistance $\mathrm{F_w}$. At this moment $\mathrm{D_1} = \mathrm{D}$.

$$\frac{T\cdot G_1\cdot G_\alpha\cdot\mu}{r} = f\cdot cos\alpha + sin\alpha \qquad (2)$$

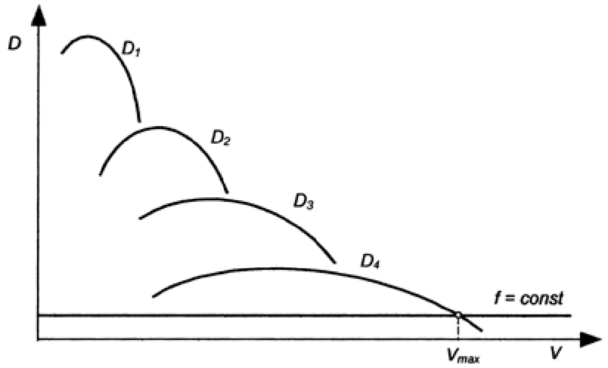


Fig.20. Dynamic index vs velocity graph for a car with 4-speed gearbox.

Let Ω be the constant value for ω.

$$\Omega = \frac{T\cdot G_1\cdot G_\alpha\cdot\mu}{r} \qquad (3)$$

For each vehicle ω, the maximal inclination angle $\alpha_{\max}$ is defined as a solution of the equation

$$\Omega = f\cdot cos\alpha + sin\alpha \qquad (4)$$

for over interval $0 \le \alpha \le \frac{\pi}{2}$.

Vehicle ω tuple can be further simplified. For example, class of simulation objects similar to a car can possess constant values: $\mathrm{G_1} = 3.909, \mathrm{G_a} = 3.353, \mu = 0.84$. Therefore, in this case, Ω can be estimated initially as:

$$\Omega = 11\cdot\frac{T}{r} \qquad (5)$$

In case when a vehicle is moving with maximal velocity v in a difficult terrain (described by inclined plane), it is consuming all available power P. At this moment, the thrust compensates the resistance and the vehicle is not capable of accelerating as it is moving with maximal and constant velocity. Consequently, there is no inertia. After transformations of widely-known formulas, velocity of a car at the time of upward movement is a solution of the following equation:

$$v_a\left(mgf\cos\alpha - mg + \frac{1}{2}C_x\rho kwh\cdot v_\alpha^2\right) - P=0 \qquad (6)$$

6 Problem Solution

Terrain translocation strategy enforces usage of algorithm that is adapted to large sets of terrain metadata. This the reason to modify well-known algorithms for solving the shortest path problem [1]. The movement algorithm implementation has been inspired by merging higher-level squares in b-nodes and decomposition [16]. The top-down method has been applied on the basis of a multiresolution grid. This approach is an effective approach to keep a balance between memory consumption and algorithm execution time. The algorithm has been divided into three essential stages:

- finding a shortest path at a given scale,
- search for the best possible route between model nodes (square areas) at the higher scale,
- complementing the path with routes through square borders.

After assigning the displacement task, a process of metadata extraction is being performed for squares in the area of interest at minimal possible scale. During a pre-filtering stage, the loaded sub-grid adheres to the above-described terrain resistance model. This approach (depending on the terrain data and properties of a simulation object) gives possibility of data size reduction thus lowering the amount of work needed to be performed by lower order algorithms (ex. Dijkstra or A*). Executed preparation actions allow to execute the first step of the relevant algorithm using A*.

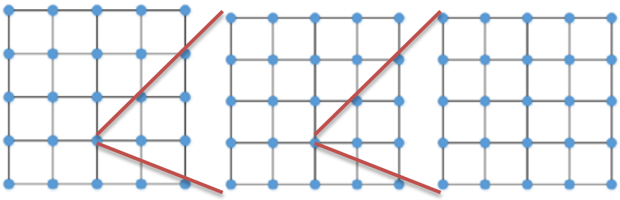


Fig.21. SymSG terrain model multiresolution grid.

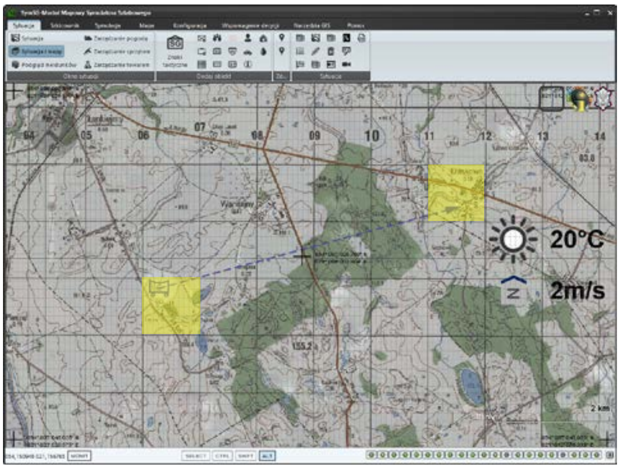


Fig.22. Visualisation of BG displacement task in SymSG environment – BG patrol movment with start and finish areas (initial phase).

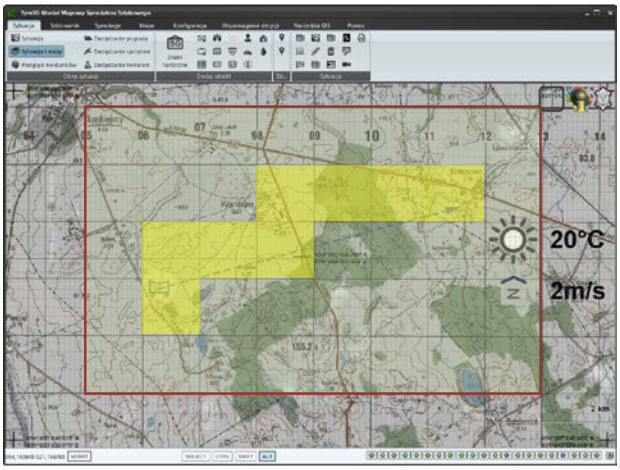


Fig.23. Visualisation of BG displacement task in SymSG environment – evaluated BG patrol movment path (final phase)

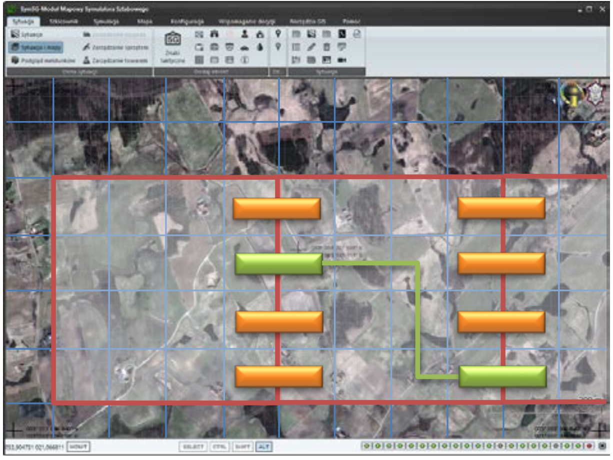


Fig.24. Evaluation and selection of the best square border routes (passages).

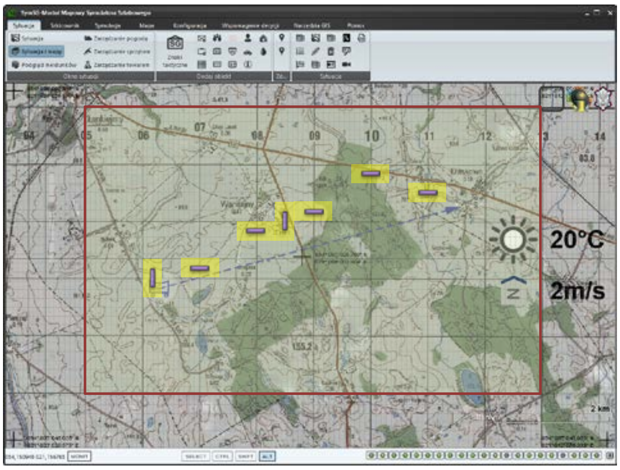


Fig.25. Selected terrain nodes - square border passages at higher map scale.

The search for the best possible square passage is the part that is responsible for pathfinding acceleration. The small size of the best border path-finding problem allows to perform exhaustive analysis of each pair of squares and their border's passage cost. The applied function evaluates a border route (passage) cost as a cost of directed edge between vertices representing adjacent terrain nodes (squares) at higher scale. Extraction of border passages allows to complement the path with coordinates of semi-origins and semi-destinations. After this step, current scale is being increased and procedure is restarted for decomposed displacement tasks.

See [20] for more information on map rendering and time complexity comparison with other algorithms, such as A*.

```
shortestPath = path.create(origin,destination);
for(int level = minLevel; level <= maxLevel;
 level++)
{
  shortestPath = shortestPath.find(level);
  if(level<maxLevel){
    foreach(adjacentSq in shortestPath)
    {
      bestBorderPassage = adjacentSquaresPair
        .findBestBorderPassage(level++);
      shortestPath.insertPathBetween(adjacentSq,
        bestBorderPassage);
    }
  }
}
return shortestpath;
```

Fig.26. Core implementation of multiresolution movement algorithm

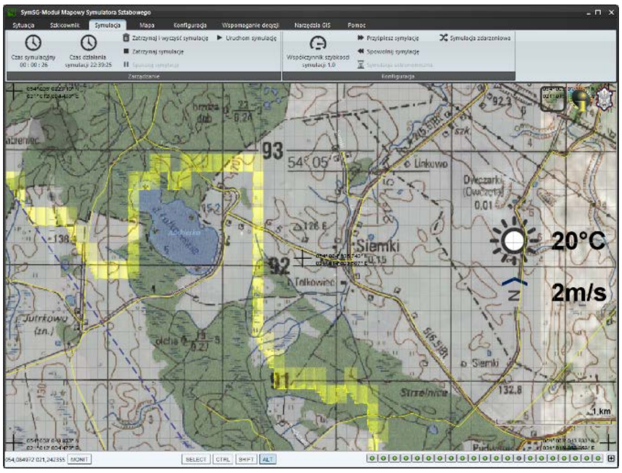


Fig.27. Visualisation of movement task execution for BG operational element using TTS algorithm – map scale 1:50 000 CADRG tiles (terrain grid).

7 Conclusion

The paper described methodology connected with obtaining some large areas of high resolution terrain, calculated on OpenStreetMap resources and serialising it as SymSG simulator's usable format. Ultimately SymSG delivers detailed discreet terrain model, which takes into account many terrain characteristics, ready to be used for simulation of various activities (reconnaissance, camouflage, pursuit) and processes (e.g. society process simulation). As presented, SymSG development showed that there is still a space for improvement in OSM.

Modelling the BG activities and requirements for highly sensitive movement processes imposed specific characteristics of algorithms. Inside the paper we have showed the algorithm that may be employed to find a path between two points of a large gridded area of diversified terrain in a reasonable computational time. Proposed set of path finding and movement algorithms include detailed terrain model, rich set of aggregated characteristics (coverage factors: forests, water, swamps, roads, infrastructure, urban areas, ducts, etc.), seasons, weather, soil types, as well as equipment, vehicle and unit efficiency.

The developed simulation environment, provides a set of effective simulation algorithms and interactive tools for training Border Guard officers within CAX training methodology. It reflects actions related to the protection of the EU and Schengen external borders.

The current solution is an effect of BG knowledge acquisition and real-world scenario data taken from CSSG experts supporting the project. The environment architecture is extendable and open to external tools integration, thus offering a large room for further work and extensions.

Acknowledgments: this work has been supported by the National Centre for Research and Development grant DOBR/0023/R/ID3/2013/03.

References:

[1] Berry J. K., *Topic 9 - Math/Stat Framework for Map Analysis (Further Reading)*,) website: http://www.innovativegis.com/basis/BeyondMapping Series/BeyondMapping_IV/Topic9/FurtherReading9.htm

[2] Border Guard Training Centre, Kętrzyn, (access: 2017) website: http://www.cs.strazgraniczna.pl/

[3] Chmielewski M., 2009, *Ontology Applications for Achieving Situation Awareness in Military Decision Support Systems*, ICCCI 2009, LNCS, vol. 5796, pp. 528-539, DOI: 10.1007/978-3-642-04441-0_46

[4] Chmielewski M., et.al., (Chmielewski) 2014, *Development of Operational Picture in DSS Using Distributed SOA Based Environment, Tactical Networks and Handhelds; ISAT: Selected Aspects of Communication and Computational Systems*, Wroclaw, ISBN/ISSN: 978-83-7493-856-3

[5] Chmielewski M., Gałka A. (2010). *Semantic Battlespace Data Mapping Using Tactical Symbology. Advances in Intelligent Information and Database Systems*, DOI: 10.1007/978-3-642-12090-9_14, 157-168.

[6] Flurry G., Clark K., *The Enterprise Service Bus, re-examined Updating concepts and terminology for an evolved technology*, 2011, IBM developerWorks (access 2017)

[7] Hoffmann R., Kiedrowicz M., and Stanik J., (2016) "*Evaluation of information safety as an element of improving the organization's safety management*", 20th International Conference on Circuits, Systems, Communications and Computers, MATEC Web of Conferences, Cedex A: E D P Sciences, vol. 76

[8] Hoffmann, R., Kiedrowicz, M., Stanik, J., 2016, *Risk management system as the basic paradigm of the information security management system in an organization*, 20th International Conference on Circuits, Systems, Communications and Computers, MATEC Web of Conferences, vol. 76.

[9] Johnson R.E., Foote B., (2014) *Designing Reusable Classes, Journal of Object-Oriented Programming*, vol. ½, Department of Computer Science University of Illinois at Urbana-Champaign. pp. 22–35

[10] Knight P., Corder A., Liedel R., Giddens J., Drake R., Jenkins C., Agarwal P. (2015), "*Evaluation of Run Time Infrastructure (RTI) Implementations*", The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology. Society for Modeling and Simulation International.

[11] Najgebauer A., Pierzchała D., Dyk M., (2015) Agent-Based M&S of Smart Sensors for Knowledge Acquisition Inside the Internet of Things and Sensor Networks, LNCS v.9012, ACIIDS pp 224-234

[12] OpenMap framework website (access: 2017), http://openmap-java.org

[13] OpenStreetMap wiki database, OpenStreetMap (access 2017) http://wiki.openstreetmap.org/wiki/

[14] Performance specification Digital Terrain Elevation Data (DTED), MIL-PRF-89020B, 23 May 2000

[15] Raster Product Format, Library of Congress, 2011, https://www.loc.gov/preservation/digital/formats/fdd/fdd000298.shtml

[16] Tarapata Z., JTIT, *Military route planning in battlefield simulation: effectiveness problems and potential solutions*, 4 47-56 (2003)

[17] VBS3 BISim wiki and VBS3 product and tools page (access 2017): https://bisimulations.com/virtual-battlespace-3

[18] OpenStreetMap wiki database, OpenStreetMap (access 2017) website: http://wiki.openstreetmap.org/wiki/Downloading_data

[19] OpenStreetMap wiki database, OpenStreetMap (access 2017) website: http://wiki.openstreetmap.org/wiki/Overpass_API

[20] Chmielewski M. et al., 2017, *High detail terrain models and multiresolution path finding algorithms for Border Guard constructive simulator. A study of effective movement algorithms in high resolution simulation environment*, Conference: 21st International Conference on Circuits, Systems, Communications and Computers (CSCC 2017), At Crete, Greece.