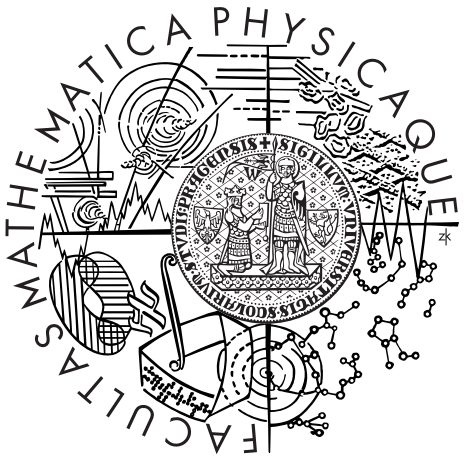
Charles University in Prague
Faculty of Mathematics and Physics

HELP



Bc. Jan Dufek

Surveillance Simulator

Prague 2014

Contents

1. Surveillance Simulator

In the following chapter the design of the GUI will presented and explained. Also all the program functions will be described.

1.1 Layout

The purpose of the GUI is to allow an easy access to all the functions of the application and to enable entering input data and visualizing the process of simulation. The GUI consists of four main parts:

1. bird view,
2. bottom bar,
3. right bar,
4. menu.

The basic layout is depicted in Figure 1.1.

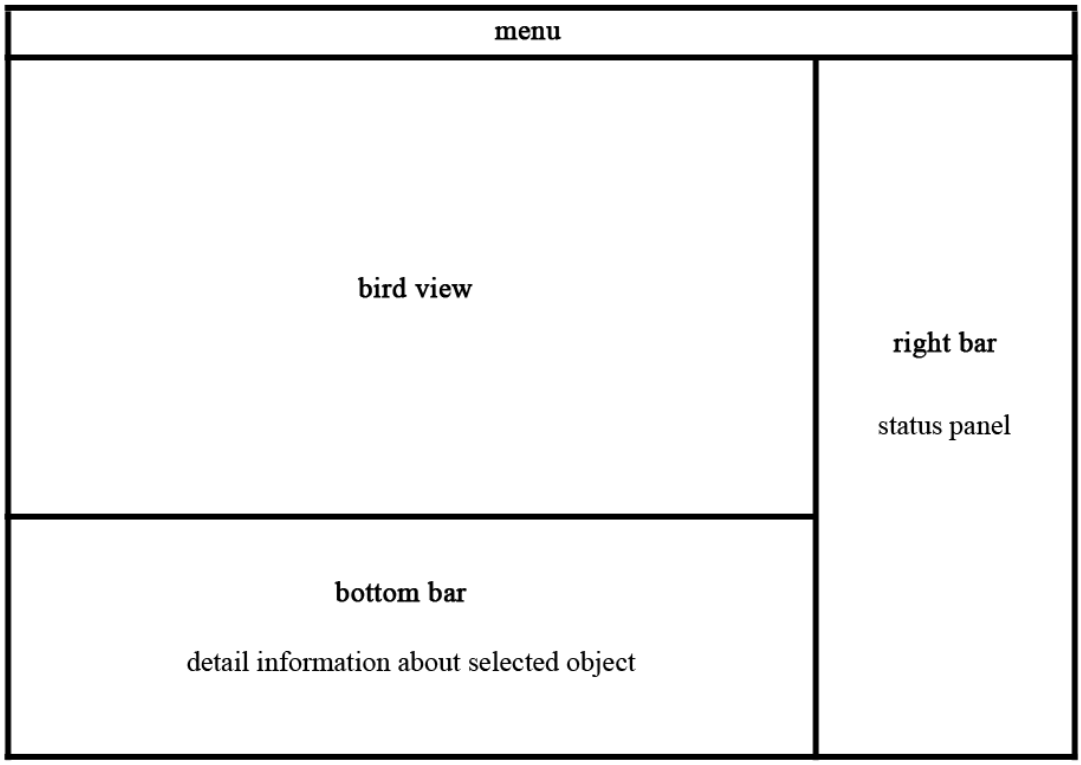


Figure 1.1: GUI basic layout.

The GUI was programmed in Java using Swing library. Final version could be seen in Figure 1.2. In the following sections all the main parts of GUI will be described.

Figure 1.2: Final form of GUI.

1.2 Bird View

The bird view is the main part of the GUI. On the background the bird view of the selected virtual environment is displayed. The bird view could be zoomed in (left double click), zoomed out (right double click) or dragged similar to web based on-line maps. The drag works also with a movement inertia (if dragged and dropped the map continues to move and stops gradually) for convenient work with the bird view.

The bird view has three main functions. The first is entering input data. The second is visualizing the progress of simulation. The third is selection and manipulation with objects.

1.2.1 Entering Input Data

After the left click anywhere on the bird view is performed, the context menu is displayed with options to add an object to the click location. There are three kinds of objects which could be added:

- UAV,
- charging station,
- area.

After clicking on the UAV or the charging station, the particular add object dialog is displayed. In this dialog various parameters of the particular object could be set. After hitting the add button, the object is added to the particular point. After clicking on the area, the polygon can be selected by clicking anywhere on the bird view. The polygon selection is confirmed with double click. Then add area dialog is displayed and the parameters could be set. The particular parameters which could be set will be now explained.

UAV

The add UAV dialog is presented in Figure 1.3. All the fields are filled with default values. Each field will be now explained. **Name** is the name of the UAV. By default the name is filled automatically with string *UAV* followed by number which is increased automatically. **Battery capacity** is the capacity in mAh of the battery on-board the UAV. **Amperage mean** is the average energy consumption in Amperes per hour. **Linear velocity** is the velocity of translation movement in horizontal plane. It is entered as the percentage of vehicle's maximum possible velocity. Maximum possible velocity depends on particular robot which is used and can be set in USARSim robot configuration file. **Rotational velocity** is set in radians per second. **Detector field of view** is set in degrees. Detector is the main sensor carried for the purpose of surveillance. The UAV is covering area using this sensor. It could be, for example, a camera, a thermal camera, etc.

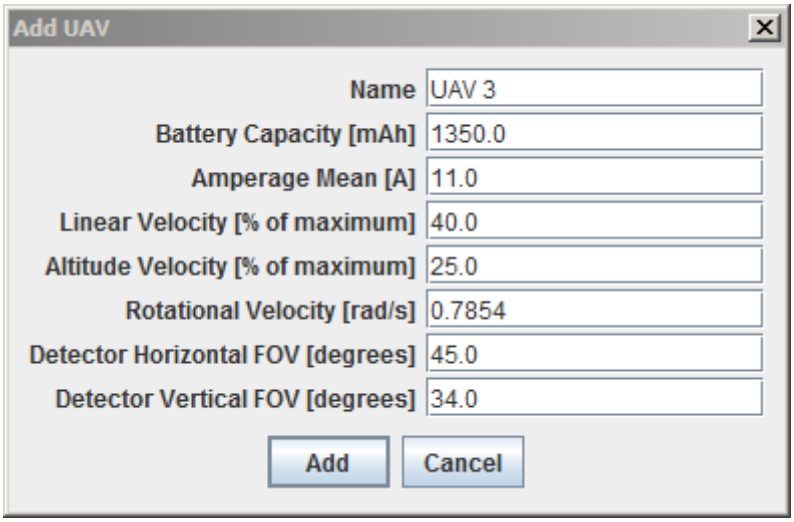


Figure 1.3: Add UAV dialog.

1.2.2 Add Charging Station

Add charging station dialog is presented in Figure 1.4. All the fields are also filled with default values including the name. Each field will be now described. **Number of batteries** is the total number of batteries in the station charging station. **Capacity of batteries** is the capacity of every single battery in the station in mAh. **Charging performance** is the performance of every single charger. There is the same number of chargers as batteries so every battery is charged with the same performance (all the batteries could be charged at the same time). **Swap time mean and deviation** are given in seconds and represents the time needed for changing the batteries between the charging station and an UAV. Because this time is not constant, the program supports to enter mean and standard deviation and then the time is simulated using normal distribution.

1.2.3 Add Area

After clicking on add area option the area selection mode will be activated. In this mode it is possible to select all the points of a polygon representing an area by clicking anywhere in the bird view. The process of selecting a polygon is illustrated in Figure 1.5. A polygon

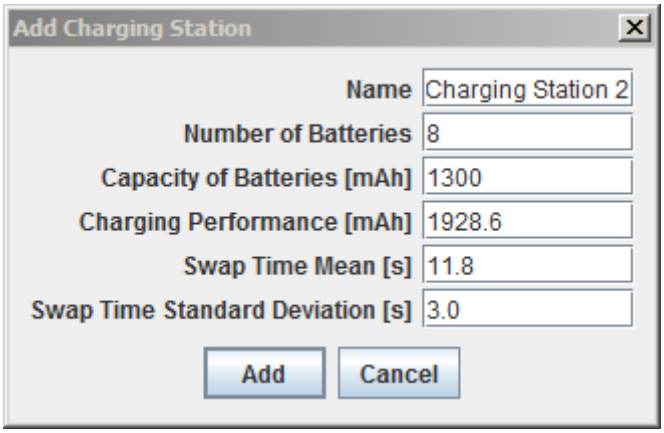


Figure 1.4: Add Charging Station dialog.

of any kind could be selected including non-convex polygons. Using multiple polygons it is possible to select any planar area. After confirming the selection the add dialog is presented. The add area dialog is shown in Figure 1.6. The only **type** of area which is supported is guarded area. It is the area which is about to be surveyed by UAVs. The other types of areas are expected to be supported in the future. For example no fly area, safe landing area, etc. The more will be discussed in future work section in conclusion chapter.

1.2.4 Visualization

The bird view is also used to visualize the process of a simulation. The first which can be seen in the bird view are the objects (UAVs, charging stations and areas) and their current status. Also the particular algorithms are allowed to print into bird view. Now the particular visualized objects will be described.

UAV

An UAV visual representation is illustrated in Figure 1.7. If an UAV is added, it is at first in the state of UAV request. That means it is not yet spawned into a virtual environment. An UAV could be spawned only if a simulation is running. After a simulation starts the UAV is spawned to an environment. UAV's visual representation changes its color according to the current status of an UAV. It is different for a request state, a takeoff state, an in air state, a charging state and a crashed state.

If we take a closer look, there are two locations displayed. The filled circle is the believed location of the UAV. The empty circle is UAV's real location. The rectangle around represents visible area. The visible area is the area which is seen by UAV's main detector. It is dynamically updated according to robot's actual height above terrain.

In addition if an UAV is flying somewhere, the destination is also visualized by a small rectangle. The example is in Figure 1.8

Charging Station

A charging station visual representation is illustrated in Figure 1.9. It changes its color according to the occupancy.

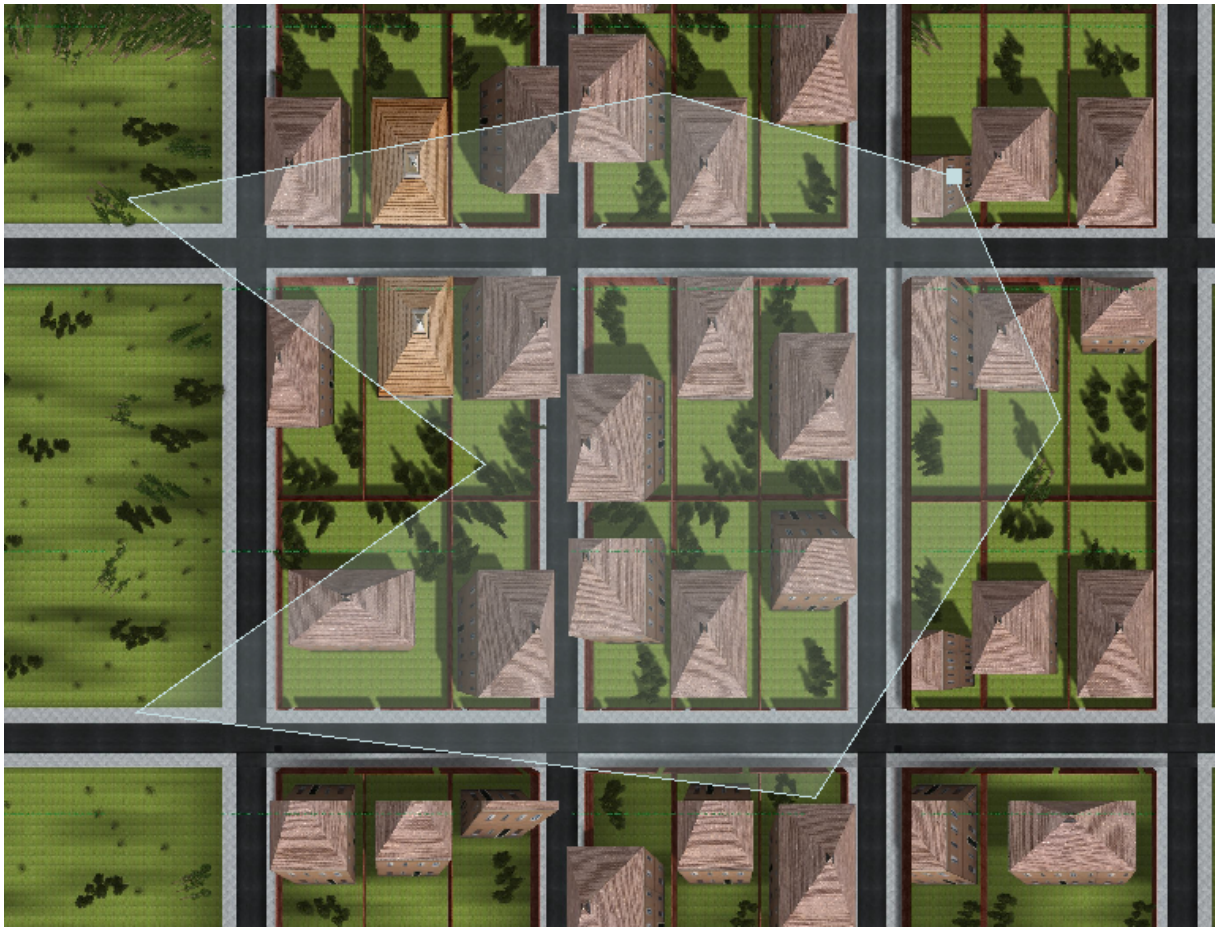


Figure 1.5: Selection of polygon.

Area

An area visual representation is illustrated in Figure 1.10. An area is composed from a bounding polygon and a heat map. Heat map visually represents the quality of coverage. If the part of an area is visible from any UAV's detector, there is a zero heat represented by green color. As long as the particular part is not visible from any UAV, the heat is rising and the color is changing towards red. The heat map will be described more in detail later.

Algorithm

Every algorithm is allowed to print anything into bird view. The information printed closely depends on the selected algorithm. The example of such visual information is illustrated in Figure 1.11. These information could be for example grids, way points, structures used by algorithms (for instance spanning trees) and so on. The particular visualization of each algorithm will be covered in the algorithms chapter.

1.2.5 Manipulation with Objects

The objects in a bird view could be selected by clicking on them. If selected, object will change its color to white. For selected object the bottom bar showing detail information is displayed. Also by right clicking on objects the editing operations could be performed

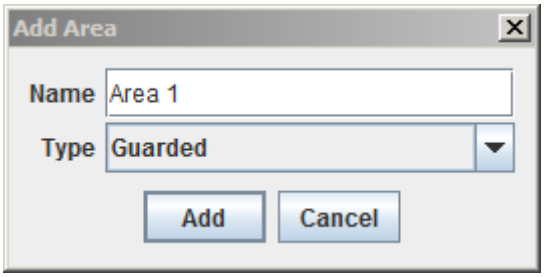


Figure 1.6: Add Area dialog.

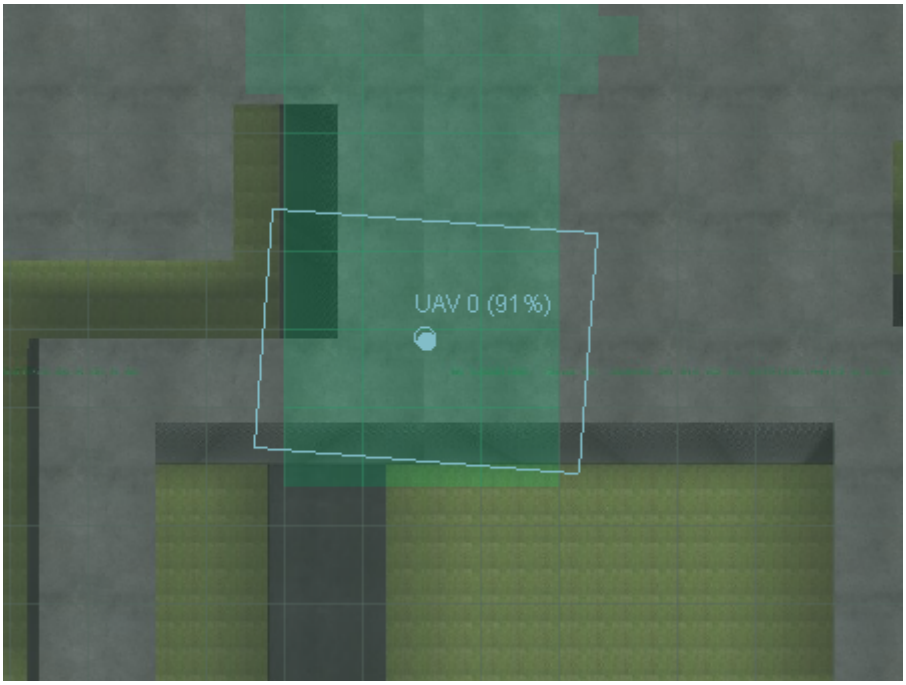


Figure 1.7: UAV visual representation.

on them. Those operations are cut, copy and delete. By right clicking anywhere on the bird view the object which is actually in the clipboard could be pasted.

1.3 Bottom Bar

The bottom bar is hidden by default. Only if some object is selected it will appear. The bottom bar could be displayed for the following objects:

- UAV,
- charging station,
- area.

The purpose of the bottom bar is to give details about selected objects. The information displayed will be described in the following subsections.

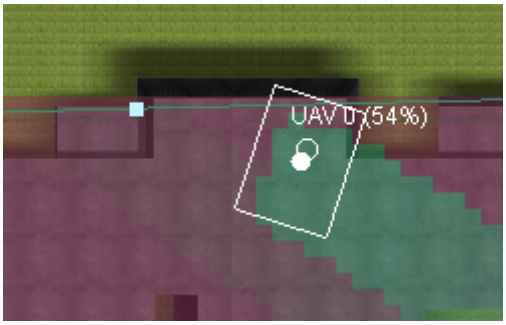


Figure 1.8: Destination visual representation.

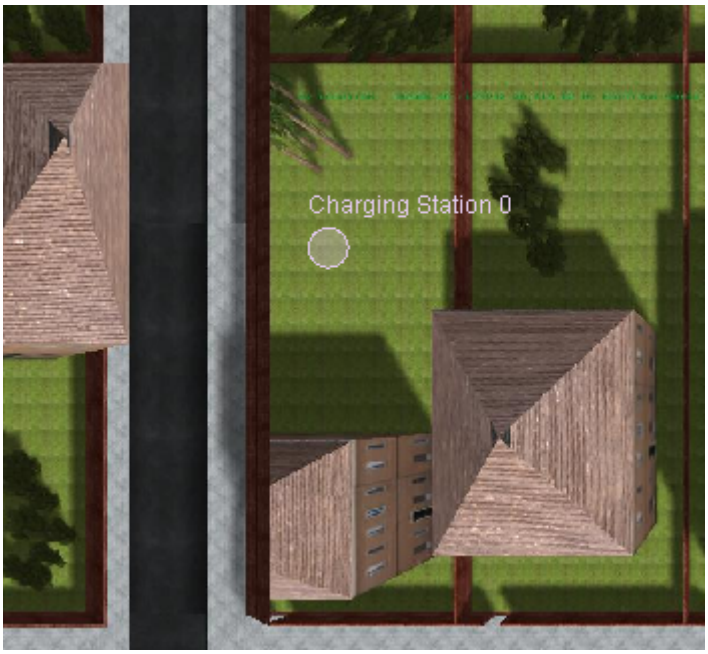


Figure 1.9: Charging station visual representation.

1.3.1 UAV

An UAV bottom bar is illustrated in Figure 1.12. All the information which were set during addition of the UAV are displayed here. Furthermore, many other real time information are shown. Now the fields of an UAV bottom bar will be clarified. **Start location** is a XYZ coordinate in USARSim coordinate system of the start point. To remind USARSim uses left-handed coordinate system and units are meters. **Start time** is the time of the server when the UAV was spawned to the environment. The time is measured as the number of seconds since the USARSim server started. **Up time** is number of seconds the UAV is present in the environment. **State** is the current status of the vehicle. State can be one of the following:

Take Off if the UAV is rising.

In Air if the UAV is flying in the air.

Landed if the UAV is landed on the ground or charging station.

Crashed if the battery died and the UAV fell down.

Figure 1.10: Area visual representation.

Actual altitude is the height above terrain in meters measured with laser scanner. **Actual rotation** is the rotation of the body of the UAV relative to the virtual environment. The rotation has three attributes. The first is pith, the second is yaw and the last is roll. Directions of these rotations were illustrated in Figure **??**. The unit of rotation is radian. **Actual and real location** are believed respectively true (real) location in left-handed USARSim coordinate system. **Deviation from real location** is difference between the actual and the real location. The higher the deviation the higher localization error.

Actual Velocity is the current speed of the the vehicle in meters per second.

Battery indicator visualizes the battery inside the UAV. It graphically and textually shows actual capacity out of the maximum possible capacity. Then it prints the percentage of battery capacity available. **Empty time** is the time in seconds when battery will die. The time is measured, as was already mentioned, as the number of seconds from the start of an USARSim server. **Remaining time** is the number of seconds till battery will die. **Number of changes** indicates how many times the UAV swapped battery in any charging station.

Charging time is the interval when the UAV is scheduled for charging. **Charging station** is the name of the charging station where the UAV is scheduled for charging. **Distance to charging station** is a distance in meters to the charging station where the UAV is scheduled. **Flying time to station** is number of seconds it takes the UAV to move above the charging station where it is scheduled. The time includes rotational time as well as translation time.

Visible area width is the width of the area seen by primary sensor which is used for surveillance. **Visible area height** is the height of this area. Both are dynamically

Figure 1.11: Algorithm visual representation.

computed according to the actual altitude.

Objectives is a list of all the objectives which are planned to be executed. Objectives are executed one by one from the first to the last. Objective system will be described in detail in the UAV control system chapter. Only certain number of objectives is displayed (the number can be set up). If there is more objectives, the note how many more is displayed below the list.

Sonars represents a sonar array of the UAV. Number of sonars and their positions on the vehicle could be set in USARSim configuration file. L denotes left sonars, M is the central sonar and R are right sonars. Sonars have the default range of 5 m. The number displayed on each sonar denotes how far an obstacle is in the direction of a given sonar. The maximum denotes that no obstacle is seen within the range. If there is any obstacle closer than the maximum range, the visual bar will be lower depending on the distance. It also changes its color from green to red according to how far the obstacle is.

1.3.2 Charging Station

A charging station bottom bar is illustrated in Figure 1.13. All the information set during addition of the charging station are displayed here. Furthermore, several other real time information are displayed. **Occupied** denotes whether the charging station is used by any UAV. This means if there is any swapping in progress. **Location** is USARSim location of the station.

Battery drum is the visualization of all the batteries in charging station. Each

battery is represented by a small bar. The capacity is indicated both visually (how much of bar is filled with color) and textually. If the batteries are under critical threshold the color changes from green to red. The batteries are sorted according to their available capacity. All batteries in a station are charged simultaneously and the charging process is visible by increase in actual batteries capacity. There is also the actual capacity percentage in brackets.

Schedule is the visualization of the schedule of the charging station. Each charging station has its own schedule to prevent collisions, because charging stations can serve only one UAV at a time. Details about charging stations' scheduling will be explained later. The schedule is visualized as a time line. On the left hand side there is the actual time which is also printed below. On the right hand side there is the end time of the last scheduled charging event. Between those two time points all the scheduled charging tasks are visualized. The graphical representation of one charging event is a color bar between two time points denoting start and end of that charging task. Charging tasks are colored with the alternating colors to visually distinguish two tasks which are right next to each other. Each charging task has also textual information on them. From top it is the name of the UAV which is scheduled, the start time and the end time. The scale of the time line is changing dynamically according to the time between current time and the end of the last charging task. Tasks are shifting towards actual time (beginning of the time line) as time is running.

1.3.3 Area

An area bottom bar is illustrated in Figure 1.14. All the information set during addition of an area are displayed here. Furthermore, several other real time information are displayed. **Area** is the area of selected area in square meters. The term *area* is there used in both meanings. The first is a quantity that express the extent of a shape in plane. The second meaning is the actual shape in plane. **Actual coverage** is the measure of the quality of coverage. It is the average number of seconds indicating how long the particular spot was not seen by any UAV. Details including how is it computed will be described later.

Actual coverage section is the visual representation of actual coverage. It shows a bar which is filled with color. More it is filled the higher is the average non visit rate and, hence, worse is the coverage. The color changes from green (the best) to red (the worst). There is the maximum threshold for actual coverage which can be set in user's settings. Above this threshold the bar's color is red and is not changing anymore. Also the bar is fully filled even though coverage numerical value may rise. This is technical solution, because otherwise the upper limit would be infinite and it would not be possible to determine the color or percentage. However, despite the limit the actual numerical value may rise towards infinity, which is desirable.

1.4 Right Bar

The right bar is visible all the time. Its main purpose is to display actual information about a simulation. The right bar could be seen in Figure 1.15. It is divided into six parts:

1. area coverage graph,
2. simulation,

3. algorithm,
4. UAVs,
5. charging stations,
6. areas.

Each part will be now described in detail.

1.4.1 Area Coverage Graph

The first part is the area coverage graph. It displays the history of coverage. On the horizontal line there is the time. On the left side of the time axis there is the start of a simulation and on the right there is the actual time. I remind that time is measured as the number of seconds from the start of USARSim server and not from the simulation start. On the vertical line there is a coverage measured as the average of spot review time for all areas. It is possible after right click to save, print, copy or zoom the graph. The graph is crucial part of algorithm analysis because it gives us the time development of the coverage. The graph was created using JFreeChart [**?**] Java library. It is a free library which makes it easy for developers to create charts.

1.4.2 Simulation

The simulation section displays general information about a simulation. **Map** is the name of the map (virtual environment) which is actually used. That is the map from which the bird view was created and which is used by server to which the application is connected. **Time** is the actual time on the server.

1.4.3 Algorithm

The algorithm section displays information about an actual algorithm used in a simulation. **Algorithm** field has three parts. The first is algorithm's name. The second is either *Multiple* if UAVs divide area between each other or *Single* if all UAVs share the whole area. The third is either *Static* if paths do not change over time or *Dynamic* if paths could change in order to maintain unpredictability which could be required by certain applications. **Actual coverage** is the actual numerical coverage right now.

1.4.4 UAVs

The UAVs section displays basic information about each UAV. The first is UAV's name. The second is actual UAV's objective. UAVs are alphabetically sorted according to their name.

1.4.5 Charging stations

The charging stations section shows basic informations about all the charging stations. The first is the name of the charging station. The second is the name of the first UAV which is scheduled for charging on this particular charging station. The third is time of start of this charging. The fourth is the available capacity of the first battery in the stack.

Batteries are sorted according to descending available capacity and the UAV gets the first battery (the one with highest available capacity). The fifth is indicating if the charging station is occupied or not (that means if the battery swap is in progress). The sixth is a total available capacity. It is the sum of all the free capacity of all batteries in the station.

1.4.6 Areas

The areas section displays basic information about areas. The first column is the name. The second is the type of the area (*Guarded* is the only supported type so far). The third is the total area in square meters. The fourth is the actual coverage of the area.

1.5 Menu

All the functions of the application which were not described in the previous sections are available through the top menu. Each part of the menu will be described in the following section.

1.5.1 File

The file menu provides an easy way to work with the files. One file contains all the objects added to the bird view (UAVs, charging stations, areas) together with all their locations and parameters. To conclude, it contains complete input data for the simulator. The support of files makes it easier to work with the simulator. User could save the actual input configuration to the file and use it later. The saved files could also be shared among users. Furthermore, if a user is using the same set of input configurations frequently he/she does not need to enter input data over and over and just open previously saved file instead.

Now the options of the file menu will be clarified. **New** will create a new file. **Open** will open previously saved file. **Save** will save the opened file to the file from which it was opened. If it is a new file it will ask for path and name where to save the file. **Save as** will save the actual file to another file under different name. It is useful when the user does not want to override the original file. Technically the manipulations with files are implemented as serialization of Java objects.

Print will print the whole bird view together with all the objects and their visual representation. Basically it will print everything which is seen by user in the bird view. It could be printed using a printer and/or saved as a PDF according to the system possibilities. **Exit** will safely exit the application. That means that if there are any unsaved changes it will ask to save them. Also any unsaved changes are indicated by the asterisk new to file's name in the title bar.

1.5.2 Edit

The edit menu provides set of tools for working with objects (UAVs, charging stations, areas). It makes manipulation with objects more convenient.

The options of the edit menu will now be described. **Clear** will clear the bird view from all the objects. The difference from creating a new file is that clear option do not create a new file but works with the actually opened file. **Undo** will undo the last operation made on the objects. There is the complete history of actions so the undo could be done

all the way to the creation or the opening of a file. The operations with objects consists of addition and deletion. **Redo** will redo the last action which was undone. **Cut** will cut the selected object. That means it will copy the object to the clipboard and then delete it. **Copy** will copy selected object to the clipboard. Note this is not the clipboard from the operating system. The simulator has its own clipboard capable of storing these objects. **Delete** will delete the selected object.

If there is any object in the clipboard added either by using cut or copy, the bird view right click context menu option named **paste** will become available. It will add the object from clipboard to the particular location where the click was performed.

The cut, copy and delete options are also available after right click on the object directly in the bird view.

1.5.3 Map

The map menu provides tools for manipulation with a map (meaning a virtual environment). It is also the place where the bird view creation utility described in chapter **??** could be accessed.

The options of the map menu will now be explained. **Change** option can be used to change the current map (virtual environment). The dialog asking for the map name will be displayed. The map name must exactly match the name of the map file in UT2004 installation directory in *Maps* folder without the file extension. The application automatically checks if the bird view exists for the selected map. If yes, it will load it. If no, it will ask whether to create a bird view now. If users agrees the bird view creation procedure is initiated as was described in **??**. After, the created bird view is loaded.

Refresh will re-create the bird view for the current map. This feature is useful if the map file is changed somehow. For example if user wants to add something (for example a new building) directly into the map using UnrealED 3.0.

1.5.4 Server

The server menu can be used to control USARSim server. The options will now be clarified. **Start** will start a server for the given map. The dialog which will appear will ask for the map name. The current map name is filled by default in the field. If the selected map is the same as current map and bird view exists, it will start USARSim server on the current map. If the map is different, it will change the map in application and loads the bird view for given map. If the bird view does not exists it will offer user to create it. The server is started by default on local host. **Stop** will terminate the USARSim server.

This only one of the three ways to run USARSim server. The other ways will be described later.

1.5.5 Simulation

The simulation menu controls a simulation. The options are the following: **Start** will start a simulation on the server which is set in application preferences. The dialog asking for algorithm selection will be displayed. From the first combo box the algorithm can be selected. Then the *Area Division* option must be set either to *Single* if all the UAVs should share the surveyed area or *Multiple* if the UAVs should divide the area between

them. The last option in algorithm dialog is *Variability*. This could be set either to *Static* if paths of UAVs should not change or *Dynamic* if paths should change to achieve unpredictability. **Stop** will terminate the running simulation. After the termination of the simulation the bird view objects will restore to the state before the simulation started.

Certain program features are not available during the run of a simulation. These features will be automatically deactivated after a simulation has started. Those features are for example file manipulation, map options and server options.

1.5.6 Preferences

From preferences menu all the user's preferences can be set. I emphasize that only user's preferences are accessible. Programmer's preferences are not accessible through GUI and can be set only through Java source code. User's preferences are saved into a file using Java serialization and thus are loaded again after the application started. So a user does not need to set everything up after each start of the application. If no preferences file is found after the application started, the default values are used. User can have multiple preferences files and switch between them or share then with other users. By clicking on **View Preferences** the preferences dialog will be displayed. There is a button for saving and also a button for resetting to default settings. Dialog contains multiple tabs and each of them will be now described.

General

On the general tab the basic preferences can be set. **UT2004 Home Folder** is a full path to the installation folder of UT2004. **UT2004 Screen Shots Folder** is a full path to the folder where screen shots from UT2004 are saved. The location depends on operating system. For example in Microsoft Windows XP the folder is in UT2004 directory. However, in Microsoft Windows 7 the folder is in ~\AppData\Local\Virtual Store\Program Files (x86)\in UT2004 folder (the name also depends on the actual installation). **UT2004 Default Map** is the default map which is loaded including a bird view every time the application starts.

Birdview

On this tab the important settings for creation of a bird view are present. **UT2004 horizontal and vertical resolution** is the resolution of UT2004. It is by default 800 $\times$ 600 px, but it may be different if a user changed it in UT2004 settings. Note that this is not the way to set a UT2004 resolution. The actual resolution must be changed in UT2004 if needed. This option is only the information for the application what the actual resolution really is and must be set to match exactly the settings of UT2004. **UT2004 Horizontal Filed of View** is the field of view (FOV) in degrees. Same as in the case of the resolution this do not actually change the FOV, it should be set to the value which was set in UT2004. These options should remain set to default values unless anybody changed them in UT2004 settings. **Screen shot altitude** is an altitude above terrain in Unreal Units from which screen shots should be made. This is important to set correctly according to the map used. How to select the right value was discussed in **??**.

Map

Here the preferences regarding used map can be set. Most of them are related to the browsing of a bird view. They need to be set according to the used map depending on a map size and a resolution. If set incorrectly the zoom rate, for example, may be too high or too low.

Default scale is a default factor by which the bird view is magnified. **Scale in and out** are factors by which the actual scale is multiplied when zooming in (left double click) or out (right double click). **Minimal and maximal scale** are the limits for scaling to prevent the bird view to be magnified too much or to be two small. **Drag inertia speed** is factor by which the bird view is translated after dragged and drop. **Drag inertia cooldown** is a factor by which the inertia translation speed is multiplied in every step to slow the bird view movement. **Drag inertia delay** is number of milliseconds between multiplying the actual inertia translation factor with the drag inertia cooldown factor. All these settings only relate to manipulation with the bird view and has no effect on a simulation.

Server

In the server tab the server settings are made. **Server address** is the IP address of USARSim server to which the application should connect. **Server port** is an IP port on the given server on which the USARSim is listening.

Simulation

Simulation logic interval is how often the simulation logic should executed. It can be imagined as how often are decisions made. If it is too long the system would react with delays on event and if it is too short it will over load a computer. Generally it is good to set it to be in synchronization with USARSim robots logic which is by default 0.2 s. One logic cycle is sometimes referred to as *logic beat*. **Coverage analysis frequency** is how often measured in number of logic cycles the area coverage should be computed. It does not have to be computed in every logic cycle in order to save computational power.

Algorithm

Here the general preferences for algorithms which are controlling UAVs could be set. **Collision detection UAVs' altitude difference threshold** is a threshold for vertical distance between UAVs to consider them in similar altitude (higher risk of collision). If other factors are met the collision danger is reported. The mechanism of avoiding collisions will be covered later. **Is near destination distance threshold** is a threshold for the distance of UAV from its destination. If the distance is less an UAV is considered to be near its destination. The movement characteristics will change if the UAV is near its destination to allow exact maneuvering. **Approach angle difference threshold** is the threshold for angle to destination. If an UAV is near its destination the angle difference must be lower than this threshold. If not an UAV must stop linear movement and rotate to its destination. This prevents cycling around a destination.

Heat Map

In heat map tab the preferences regarding heat map are made. How the actual heat map works will be described later. For now just brief explanation will be given. The guarded area is divided into small cells and each cell has its heat value. This value is zero by default and is rising gradually. **Cell width and height** are dimensions of one heat map cell. **Is hot threshold** is the threshold for actual heat. Above this threshold the heat is considered to be hot. It will cause that the color denoting the heat in GUI is red and it is not changing color anymore. Also the heat percentage is on 100%. However, the numerical heat is not limited and may rise. The threshold is only technical solution to allow to map infinite interval of coverage to finite interval of colors and percents.

UAV

In this tab all the preferences regarding UAVs can be set. The first five fields are reserve times for charging procedure. The charging planning and control will be described later. The reserve times are intended to prevent UAV from missing a charging or underestimating the time needed for charging. These reserve times are used to either multiply the exact computed time or to add some constant to it. **Safety margin for return back to charger** is a factor by which is the time computed to be needed to return to charging station multiplied. It is used because of unpredictable factors as collision avoidance, sensor drift and so on. **Safety margin for start of charging** is a additive constant in second giving the period in which an UAV should arrive above a charging station. An UAV is allowed to arrive anytime within this interval. **Safety margin for landing and takeoff** are factors by which the computed times for landing respectively takeoff are multiplied.

To understand the next three fields the basic idea of light levels must be given. The topic, however, will be covered later in more detail. UAVs are distributed evenly to multiple flight levels. Maximum and minimum level can be set as well as distance between flight levels. Flight levels are selected according to their occupancy to balance the number of UAVs on each level. **Maximum flight level** is a maximum altitude above terrain in which UAVs are allowed to flight in order to maintain the quality of surveillance. It should be set up according to the distance of a main sensor (detector) which is used for surveillance. Also note that a map designer should count with it and make a virtual world tall enough to allow UAVs to fly in desired altitude. **Minimum flight level** is the minimum altitude where UAVs can flight. It could be set regarding the FOV of main sensor. If it is too low, then only small area will be visible. However, if it is too high there will be less flight levels and the collision of danger will be higher. **Distance between flight levels** is a distance between two flight levels which are next to each other.

Click active area size is a circular area around UAV's visual representation given in pixels where if a click is performed it is considered as click on the UAV. Its reason is to solve problem with difficulty to click on fast moving UAV.

Spawning height above spawn level is altitude above average terrain from which UAVs are spawned. An UAV is spawned in given altitude. Then it travels down since it hits the terrain. After that an UAV is considered to be spawned in an environment. It solves a problem that it is not possible to get the Z-coordinate of a ground on particular spot. Thus it is not clear what should be the start Z-coordinate of UAV. If it is underground an UAV will not spawn. If it is too hight above terrain the UAV will crash. The solution is to slowly lower UAV since it hits the ground. **Spawn finished altitude** is altitude above terrain in which an UAV is considered to be spawned, slows down and is

woken up. This should be higher than zero otherwise UAV will hit the ground too hard during spawning.

1.5.7 Help

View help will display user's documentation for the application. **About** will show information about the application such as version, author, contact and copyright.

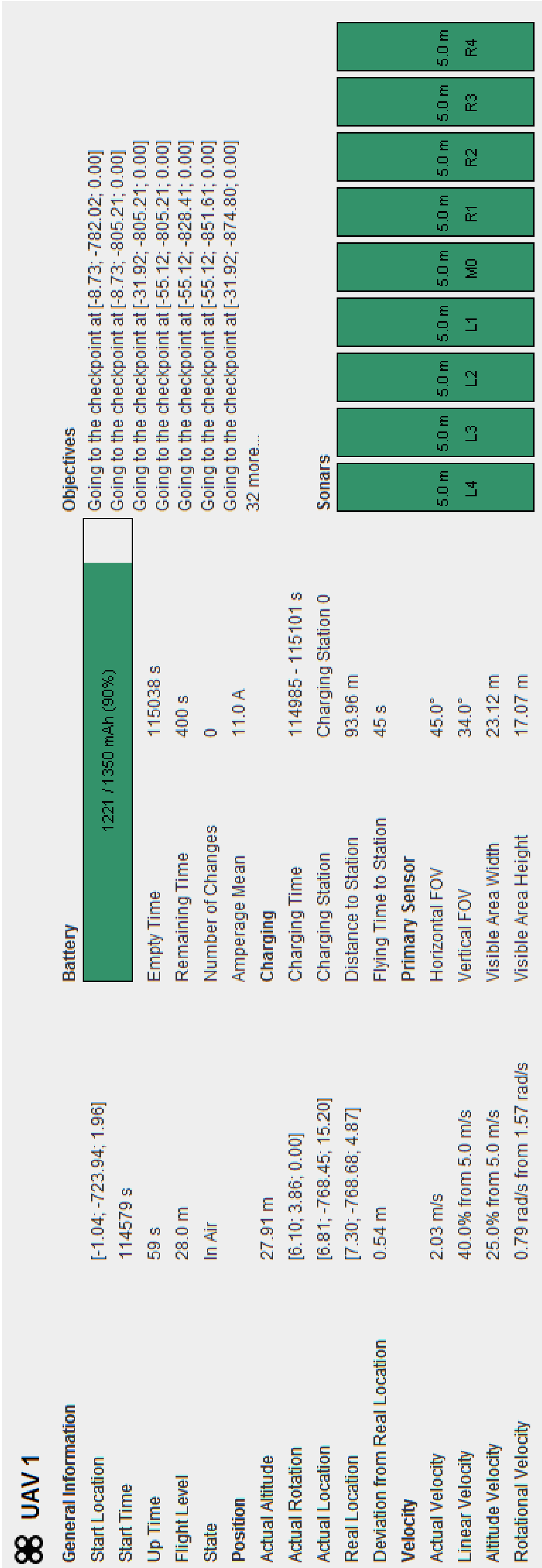


Figure 1.12: UAV bottom bar.

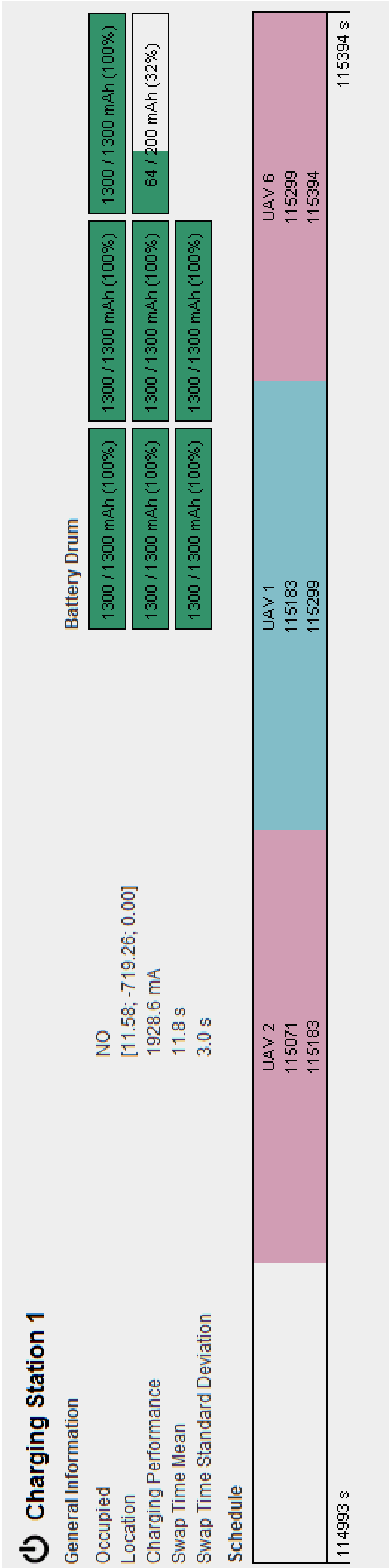


Figure 1.13: Charging station bottom bar.

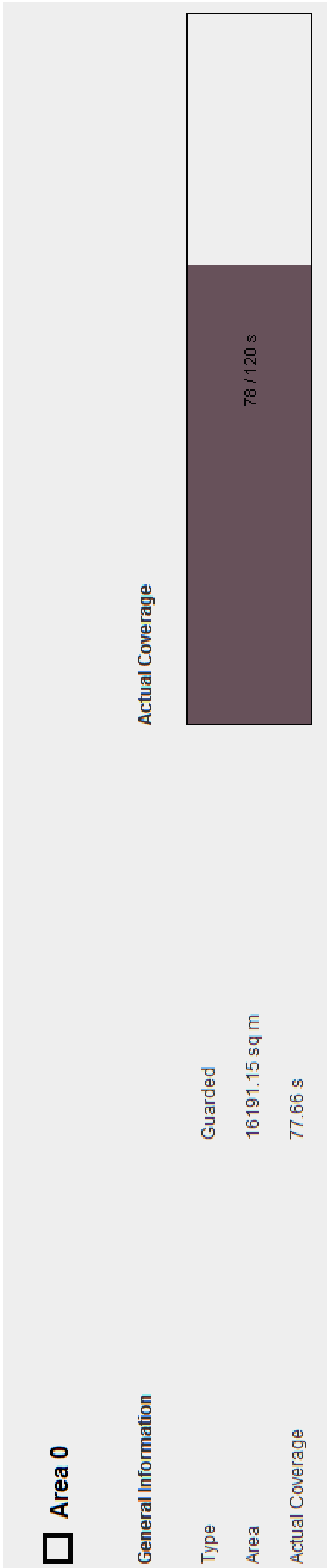


Figure 1.14: Area bottom bar.

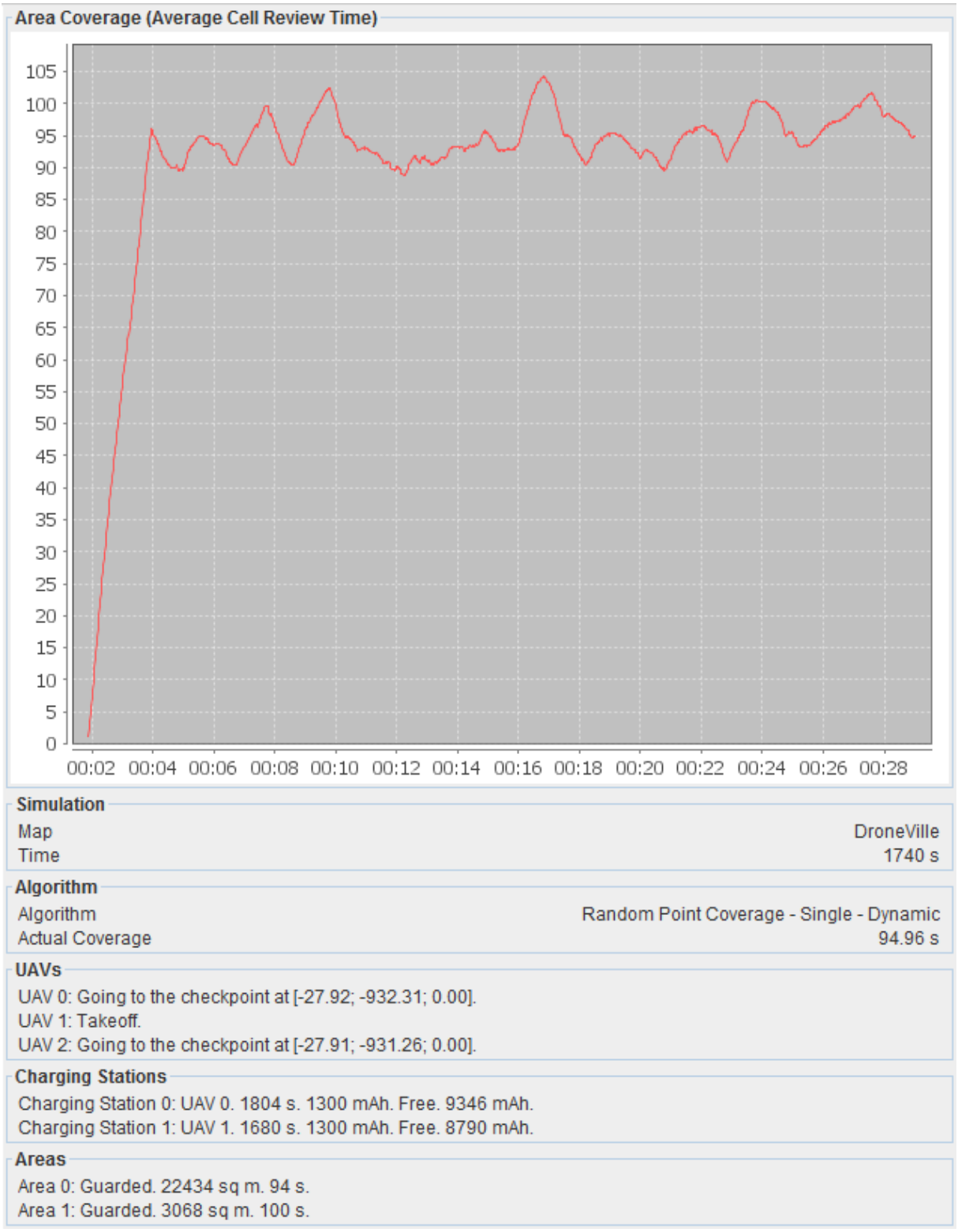


Figure 1.15: Right bar.