RoboCup Rescue Map and Scenario Creation Manual

RoboCup Rescue Simulation Team
Version 2.0, September 9, 2020

Table of Contents

1. Purpose

The RoboCup Rescue Simulator comes with some example maps that can be used to test and analyze a team of agents' performance. Although useful, these different maps do not represent all possible situations to evaluate a strategy or overall performance of a team of agents. Furthermore, teams would benefit to assess their performance in a larger number and variate of maps. Hence, it is useful to be able to create new maps for the RoboCup Rescue Simulator.

This tutorial describes the procedure to create maps and scenarios compatible with the RoboCup Rescue Simulator. The tutorial illustrates the basic steps to create a map and scenario using the creation of the map of the University of São Paulo, Brazil as example.

2. Tools

2.1. OpenStreetMap

OpenStreetMap (OSM) is a free worldwide map web platform developed collaboratively. OpenStreetMap provides free to use geographic information from many places around the world. It was created under the project *Planet OSM* aimed to provide geographic information in XML format, containing description of Nodes, Ways, and Relations. A new version of the project is released every week in a single XML file (around 93GB compressed at September 5, 2020). The entire map, or the latest weekly changeset, can be downloaded at https://planet.openstreetmap.org/. Figure 1 shows the layout of a reduced `.osm` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">
 <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900" maxlon="12.2524800"/>
 <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636"
timestamp="2008-09-21T21:37:45Z"/>
 <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter" uid="36744" visible="true" version="1" changeset="323878"
timestamp="2008-05-03T13:39:23Z"/>
  <tag k="name" v="Neu Broderstorf"/>
  <tag k="traffic_sign" v="city_limit"/>
 </node>
 ...
 <node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636"
timestamp="2008-09-21T21:37:45Z"/>
 <way id="26659127" user="Masch" uid="55988" visible="true" version="5" changeset="4142606" timestamp="2010-03-16T11:47:08Z">
  <nd ref="292403538"/>
  <nd ref="298884289"/>
  ...
  <nd ref="261728686"/>
  <tag k="highway" v="unclassified"/>
  <tag k="name" v="Pastower Straße"/>
 </way>
 <relation id="56688" user="kmvar" uid="56190" visible="true" version="28" changeset="6947637" timestamp="2011-01-12T14:23:49Z">
  <member type="node" ref="294942404" role=""/>
  <member type="node" ref="364933006" role=""/>
  <member type="way" ref="4579143" role=""/>
  ...
  <member type="node" ref="249673494" role=""/>
  <tag k="name" v="Küstenbus Linie 123"/>
  <tag k="network" v="VVW"/>
  <tag k="operator" v="Regionalverkehr Küste"/>
  <tag k="ref" v="123"/>
  <tag k="route" v="bus"/>
  <tag k="type" v="route"/>
 </relation>
 ...
</osm>
```

Figure 1. OSM Layout

2.2. Geographic Markup Language

Geographic Markup Language (GML) is a XML-based grammar used to describe interchangeable geographic information. It was defined by the Open Geospatial Consortium, and it is largely used for providing a rich set of markup primitives that allow the creation of application specific schemas, such as the definition of Buildings and Roads. The RoboCup Rescue Simulator uses the GML format for representing the entities in its maps.

2.3. Java OpenStreetMap editor

Java OpenStreetMap Editor (JOSM) is an open Java-based OpenStreetMap editor, an application originally developed by Immanuel Scholz and currently maintained by Dirk Stocker. JOSM can be used to download, edit, and convert maps from OSM to the GML format. The conversion requires a plugin that can be downloaded at RCR-converter.

2.4. `osm2gml`

The `osm2gml` enables the conversion from OSM to GML standard, transforming the XML file from one format to the other. The conversion process changes the features in the original map to make it compatible with the GML representation of the maps in the RoboCup Rescue Simulator. The `osm2gml` is part of the RoboCup Rescue Server project (see README at RoboCup Rescue Server).

3. Map and scenario creation process

In a nutshell, the process of creating a map for the RoboCup Rescue Simulator is comprised of 3 basic steps:

1. Capture the map in a OSM file format (`.osm` extension)
   The JOSM editor is used to browse through the OSM worldwide map, select the wanted area of the map, and export the map area information as an `.osm` file. Section 3.1 illustrates in detail how to perform these tasks.
2. Convert the OSM file into the GML file format
   The `osm2gml` tool is used to convert the OSM file into a GML file format. If the conversion fails, perform the map capture again using the JOSM. Section 3.2 illustrates in detail how to use the `osm2gml` tool and some recurrent changes that must be made on the OSM map before converting it.
3. Create a valid RoboCup Rescue scenario for the map on the simulator
   To use the map on the RoboCup Rescue Simulator, a scenario has to be created setting the initial position of agents, special buildings like Ambulance Center, Fire Brigade, Police Station, and Refuges. Section 3.3 illustrates in detail how to setup a scenario associated to a specific map.

3.1. Map capturing

The map capturing is performed using the JOSM tool and it is comprised of another 3 stages: install and run the JOSM tool Section 3.1.1, select and download locally a region of the map Section 3.1.2, and save it as an `.osm` file Section 3.1.3.

3.1.1. Download and run JOSM

First, download the JOSM tool from the http://josm.openstreetmap.de/. There are multiple alternatives to download JOSM, i.e., Installer, Launcher, JAR file. The recommended alternative is the JAR file, which is compatible with any environment supporting Java 8+. Once downloaded, execute the JAR file as

```
$ java -jar josm-tested.jar
```

Figure 2 illustrates the JOSM editor interface.

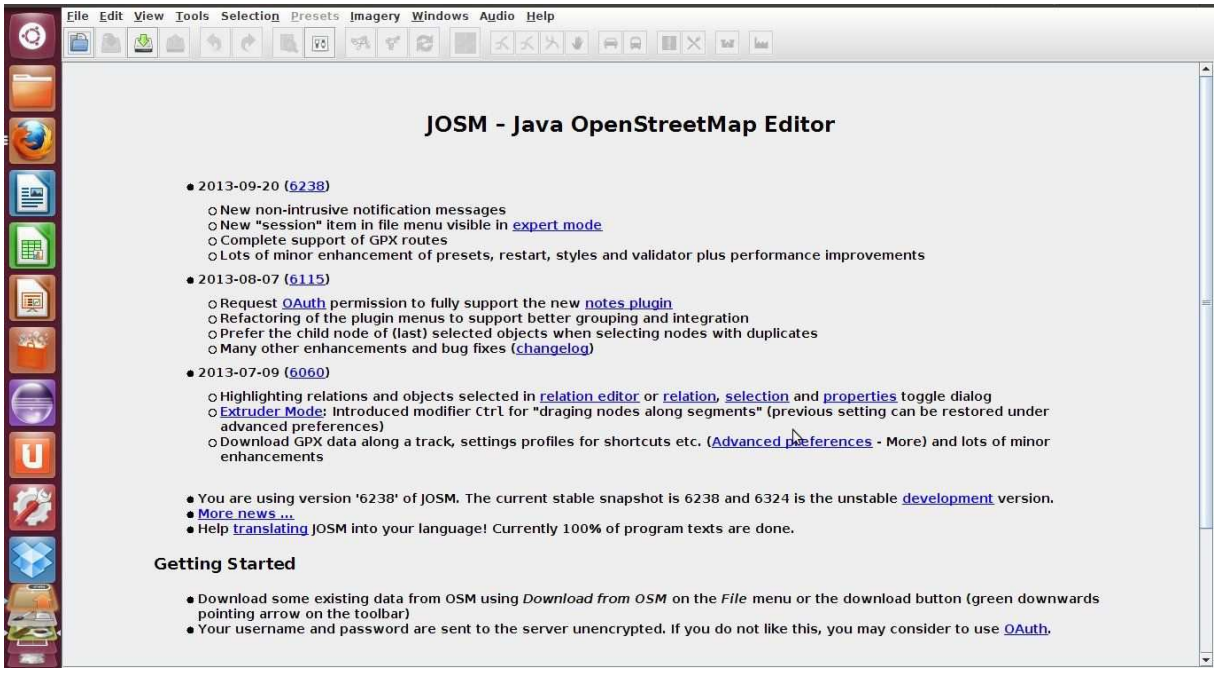

Figure 2. JOSM Editor

3.1.2. Select and download the region of the map

On the toolbar, select the green arrow button to "Download Map from OSM Server" and a zoomed out map is shown (see Figure 3).

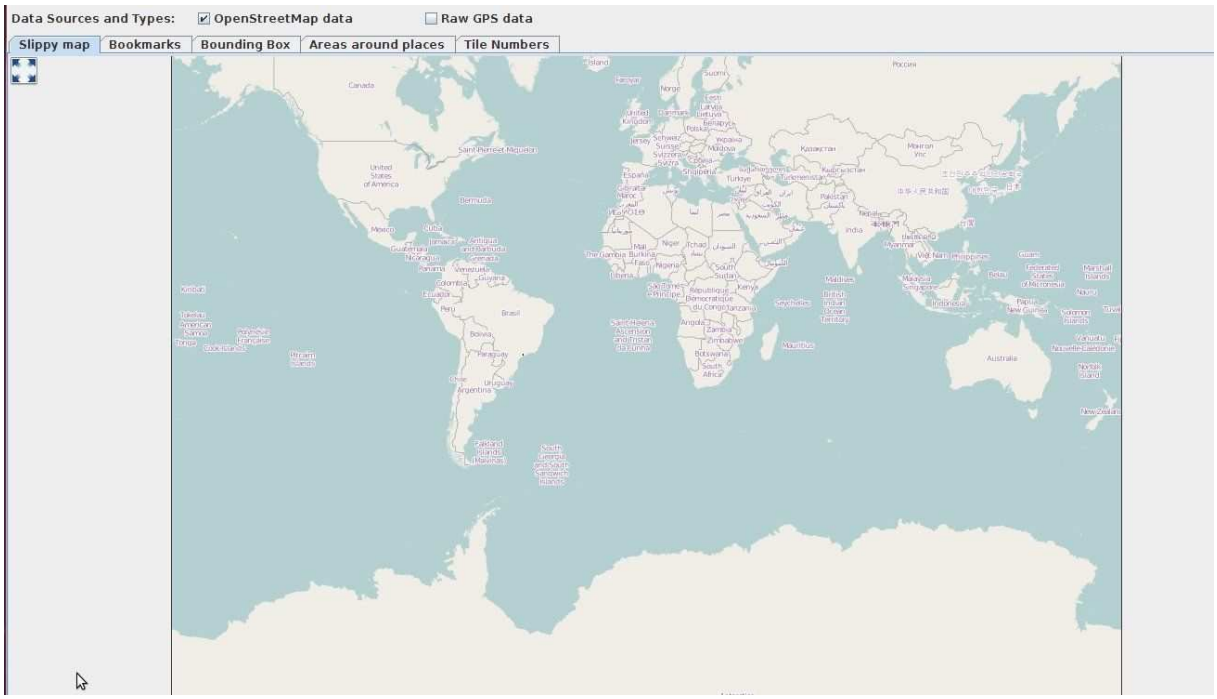

Figure 3. JOSM Zoomed Map

Manually zoom in until the region of interest is found. After finding the region of interest, select the specific area you want to download, and select the *Download* button to download the selected region. Figure 4 shows the selected area corresponding to the University of São Paulo.

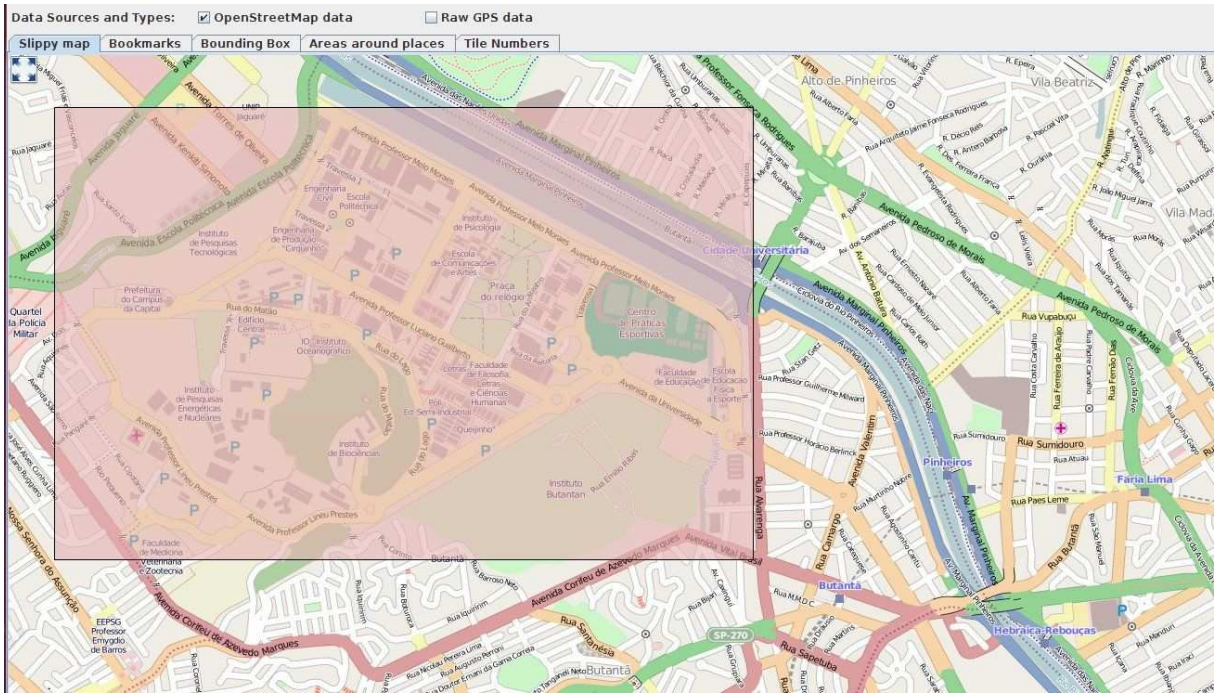

Figure 4. JOSM University of São Paulo (USP) map selection

3.1.3. Save the OSM file

After downloading the area of the map from the OpenStreetMap server, JOSM will open the edit screen with the downloaded map on display. Before starting editing it, save the map using the `File -> Save As..` menu options (see Figure 5).

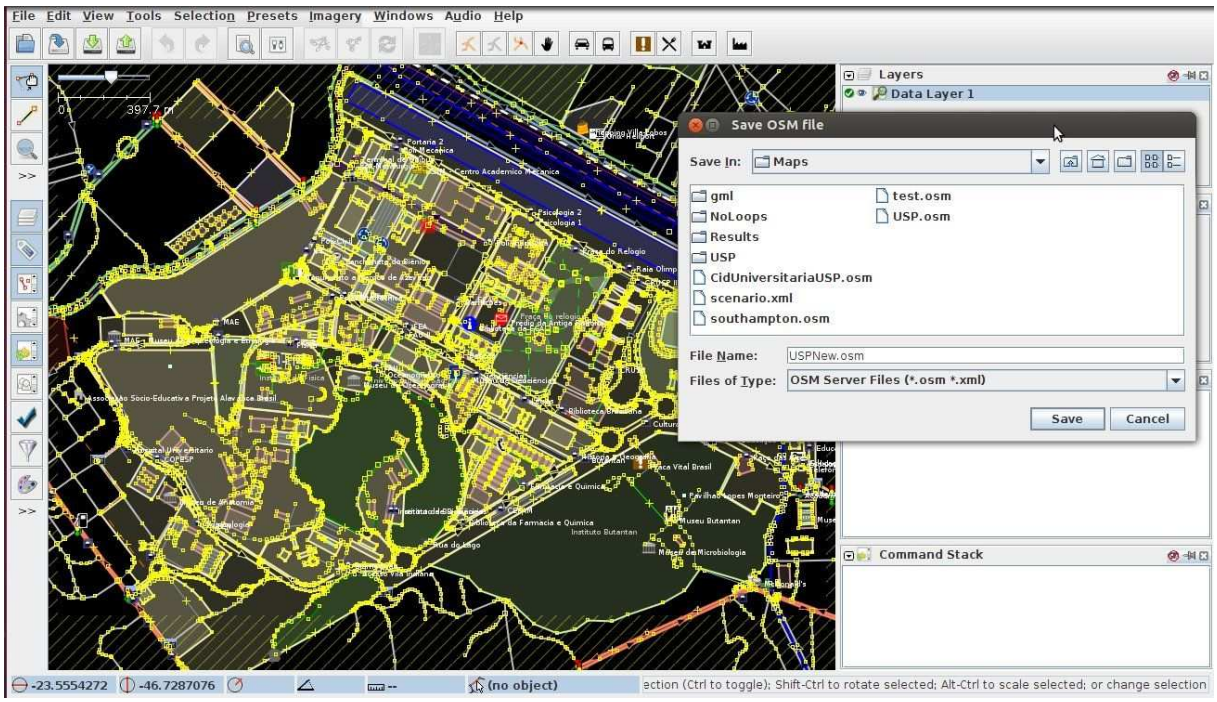

Figure 5. JOSM Downloaded University of São Paulo (USP) map

3.2. Convert OSM map into GML map format

After saving the map in OSM format, there is the need to convert it into the GML format

compatible with the RoboCup Rescue Simulator using the `osm2gml` tool. The OSM map, however, normally contains some kinds of shapes and streets that causes the conversion to fail, and some buildings and streets not marked as such, which makes them disappear in the conversion process. Section 3.2.1 describe some of these problems and how to overcome them.

3.2.1. Adjusting the OSM map

In order to make the map convertible to the GML format, some changes have to be made on the original OSM map.

NOTE	The problems reported here are not exhaustive, but purely based on experience during the creation of the University of São Paulo map. Some of these problems may not show up in other map conversions and new problems may arise.

3.2.1.1. Buildings

- Remove Buildings from outermost shapes

Some of the buildings overlap with the outermost shape in the map. The converter interprets all buildings overlapping with the outermost shape as only one, the outermost one, eliminating all buildings. To prevent this situation, it is necessary to remove the outermost shape, allowing the buildings to be processed separately. In order to remove the outermost shape, click in one of its edges, then press `Delete`. Figure 6 illustrates the case of multiple buildings inside another shape.

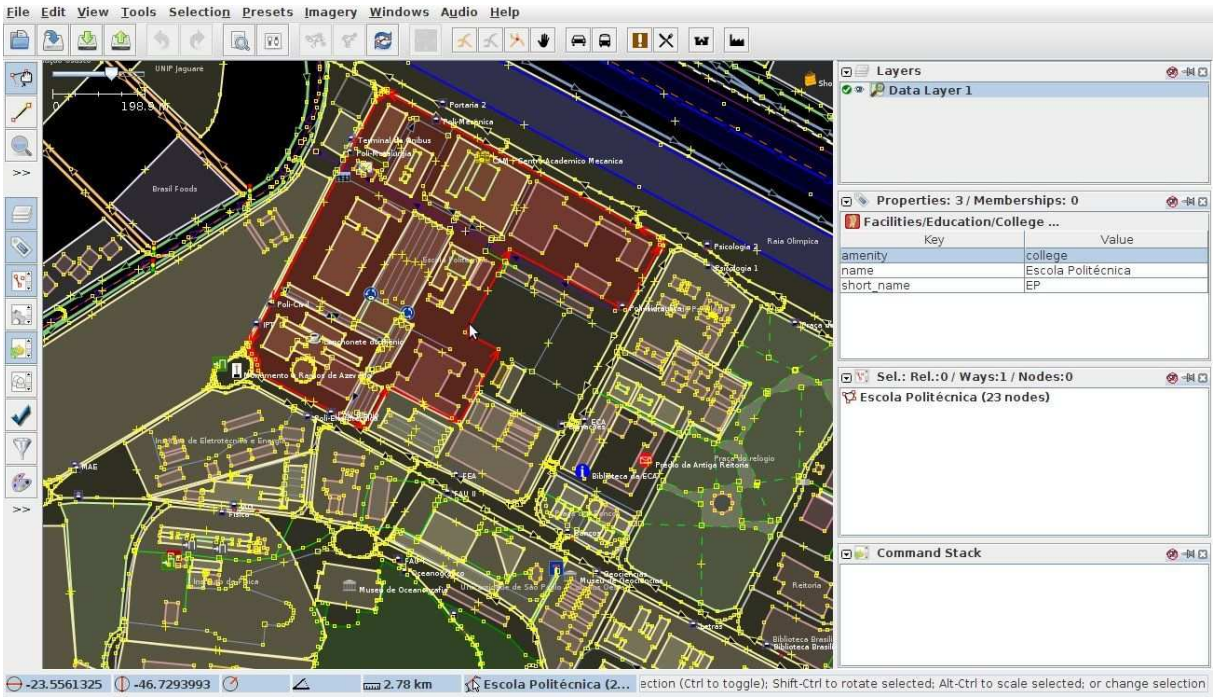

Figure 6. Outermost Shape

- Making *Buildings* as *Building*

Because most of the times shapes corresponding to *Buildings* in the original OSM map are not marked as buildings, the converter does not generate a corresponding building at the resulting GML map. Hence, it is necessary to identify manually all the non-marked buildings in the OSM map. To set one shape as Building, select the shape then go to `Presets` -> `Man Made` -> `Man`

`Made` -> `Building` menu option. Then, select one of the Building type. Figure 7 shows the setup of a shape as building.

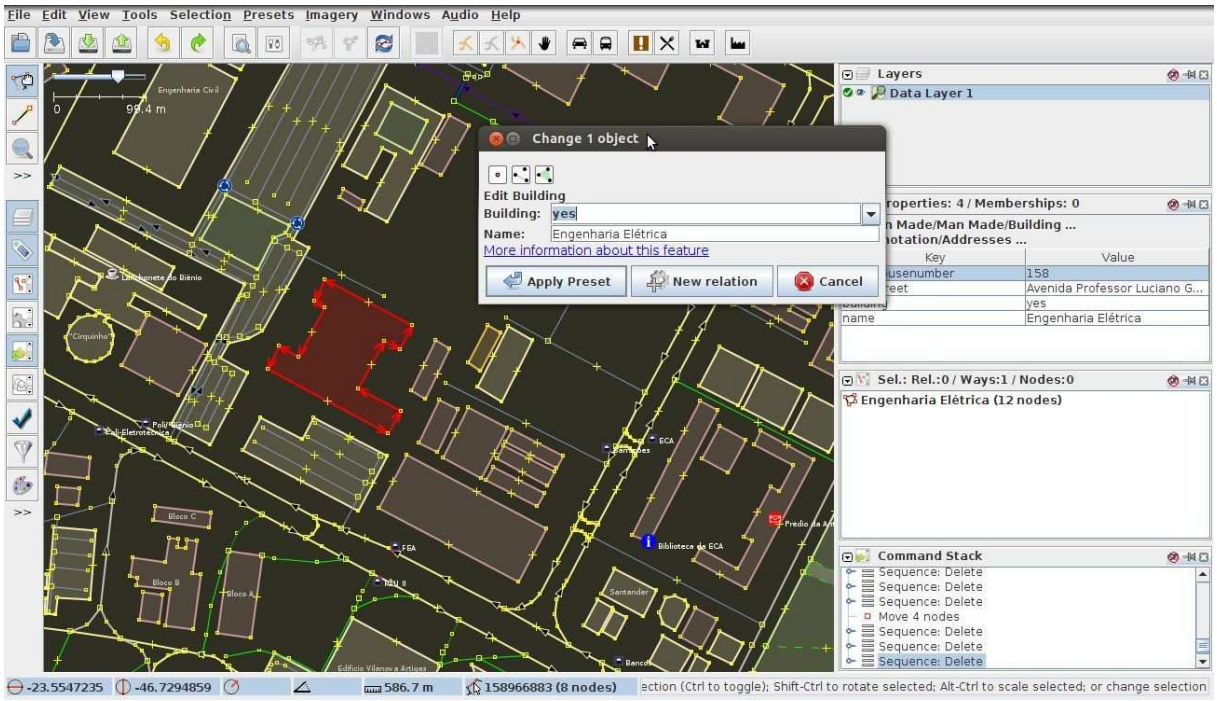

Figure 7. Building Setup

- Separate Overlapping Buildings

The original map may contain shapes that overlap each other, either two buildings, or one building and a road, or two roads. The converter processes these overlapping during the conversion process, but sometimes it fails. The safest practice is to separate the overlapping shapes in your map before converting it. Select one of the shapes and drag and drop it to separate one from the other. Figure 8 illustrates an example of overlapping buildings.

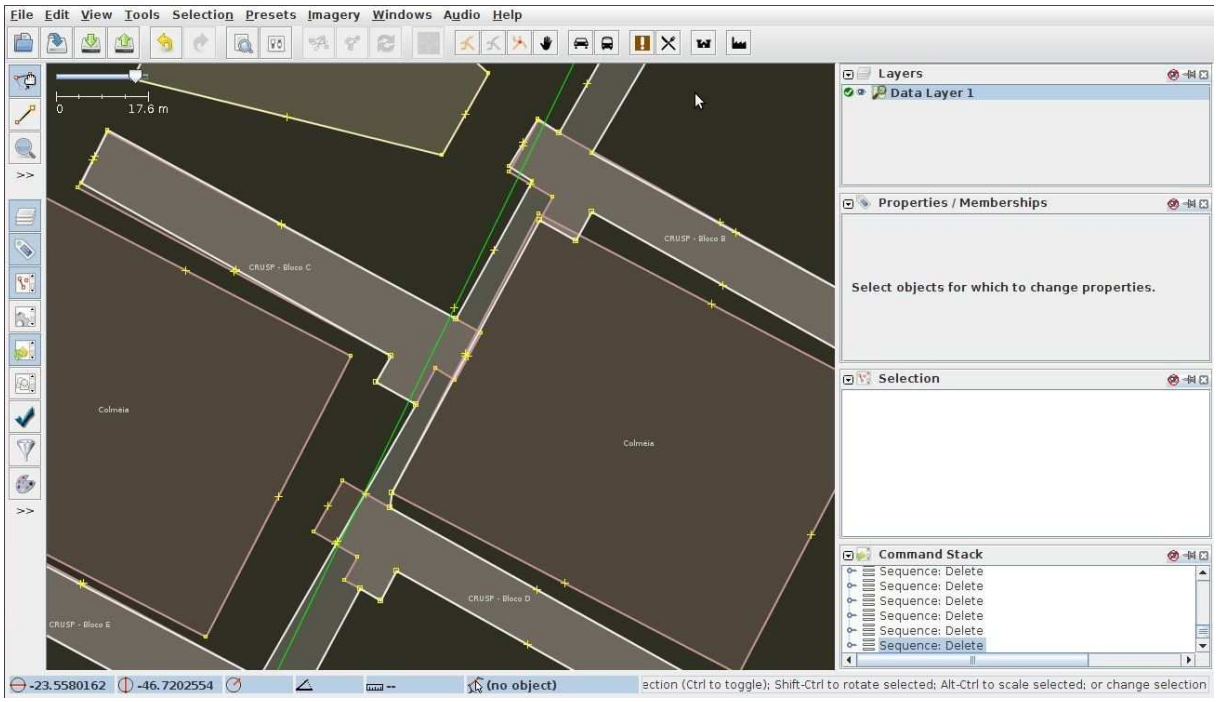

Figure 8. Overlapping Buildings

3.2.1.2. Roads

- Setup roads as both ways

Most roads on the original map are set as only one way road, although some of them should be both ways roads (see Figure 9).

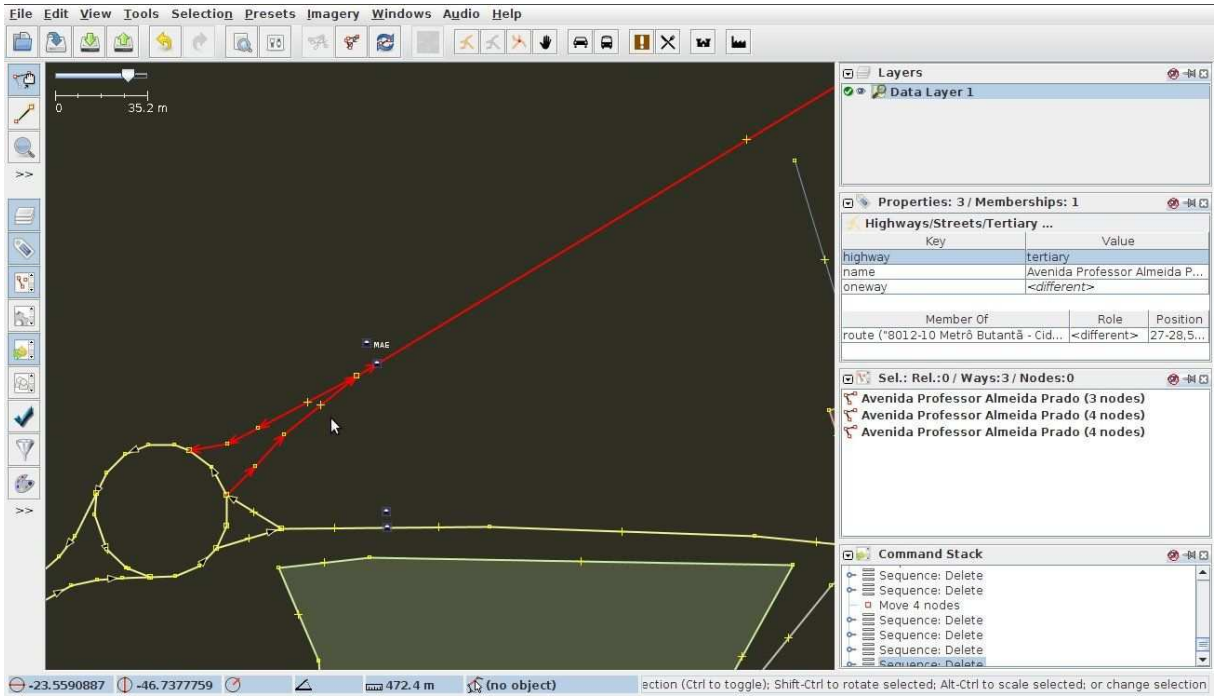

Figure 9. Both Way Road

To set roads as both ways, select it then go to `Presets` -> `Highways` → `Streets` and select the street type. A dialog will appear, before clicking `OK`, make sure the checkbox `Oneway` is not

selected. Figure 10 illustrates the dialog for the option of "Unclassified" street.

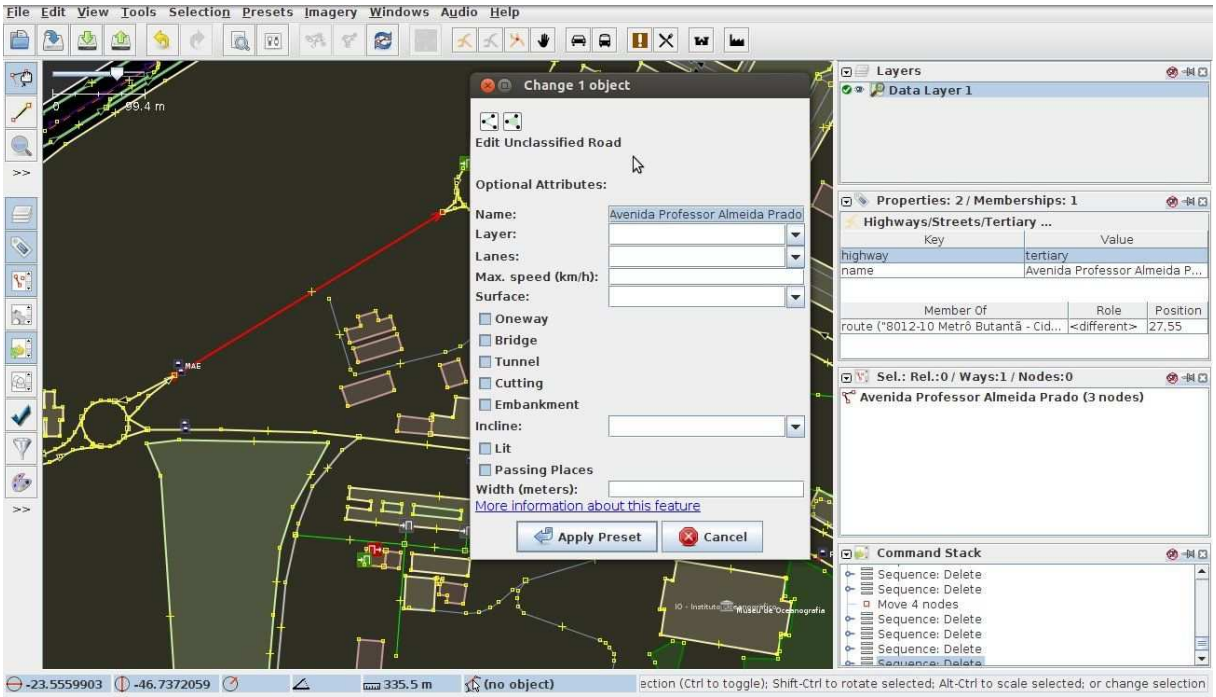

Figure 10. Road Setup Dialog

- Remove roads from inside buildings

The original OSM map may have some lines inside the buildings, which represent the path one can walk inside them. But sometimes these lines are interpreted by the converter as roads, and this can cause the conversion process to fail. To prevent this problem, it is necessary to remove these lines from inside the buildings. Figure 11 shows an example of lines inside a building.

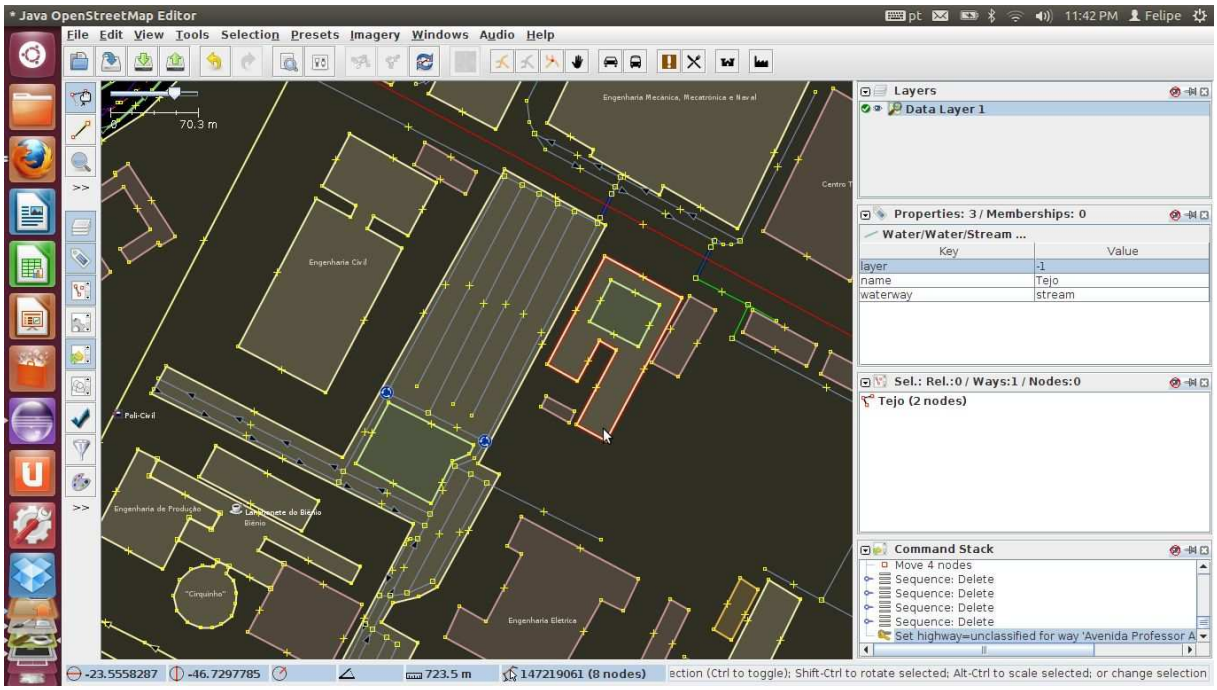

Figure 11. Lines Inside the Building

3.2.2. Running the map converter

To convert the adjusted OSM map into the GML format, it is necessary to run the `osm2gml` converter. The run the converter, open a terminal window, navigate to the `rcrs-server` root directory and execute

```
$ ./gradlew osm2gml --args='<osm map path> <gml map path>'
```

The `<osm map path>` is the path to the OSM map file and the `<gml map path>` is the destination GML map path.

> **NOTE** Even after running all the checks in Section 3.2.1, there still may be some invalid entities in the map may cause the conversion to fail. Use the JOSM to fix those newly identified error in an iterative process.

Figure 12 illustrates the converter application running, and Figure 13 shows the resulting map after the conversion.

Figure 12. Converter `osm2gml` Running

Figure 13. Conversion Result

3.3. Create a Scenario

Create a scenario means configuring the initial state of the entities in a map such as the initial location of agents, the locations of the centre agents, buildings on fire, etc.

To illustrate the creation of a scenario, the University of São Paulo created in Section 3 is used.

3.3.1. Map directory

The maps and scenarios of the current maps in the RoboCup Rescue Simulator ae stored in the directory `rcrs-server/maps/gml/`. Inside each folder in this directory there are a `map.gml` and a `scenario.xml` file. The former represents the map in GML format, while the latter represents one scenario for that specific map.

3.3.2. GML map file layout

In order to create a scenario, it is necessary to understand the layout of the GML file. The GML file is separated into 4 important parts: the node list, the edge list, the building list, and the road list.

The edge list contains information of all the edges that are composed by the nodes in the node list. The buildings and roads are defined based on the edges. Figure 14 and Figure 15 show a example of the building and road lists on a GML map file.

```
<rcr:buildinglist>
  <rcr:building gml:id="8508">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#7724"/>
      <gml:directedEdge orientation="+" xlink:href="#7723"/>
      <gml:directedEdge orientation="+" xlink:href="#3652"/>
      <gml:directedEdge orientation="-" xlink:href="#3654"/>
      <gml:directedEdge orientation="-" xlink:href="#3655"/>
      <gml:directedEdge orientation="+" xlink:href="#3651"/>
      <gml:directedEdge orientation="+" xlink:href="#7720"/>
    </gml:Face>
  </rcr:building>
  <rcr:building gml:id="8509">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#8189"/>
      <gml:directedEdge orientation="+" xlink:href="#8196"/>
      <gml:directedEdge orientation="+" xlink:href="#8195"/>
      <gml:directedEdge orientation="+" xlink:href="#4483"/>
      <gml:directedEdge orientation="-" xlink:href="#4476"/>
      <gml:directedEdge orientation="+" xlink:href="#4479"/>
      <gml:directedEdge orientation="+" xlink:href="#4474"/>
      <gml:directedEdge orientation="+" xlink:href="#8193"/>
      <gml:directedEdge orientation="+" xlink:href="#8192"/>
    </gml:Face>
  </rcr:building>
 ...
  <rcr:building gml:id="8619">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#4280"/>
      <gml:directedEdge orientation="-" xlink:href="#4282"/>
      <gml:directedEdge orientation="+" xlink:href="#4285"/>
      <gml:directedEdge orientation="+" xlink:href="#4284"/>
    </gml:Face>
  </rcr:building>
</rcr:buildinglist>
```

Figure 14. GML Building List

```
<rcr:roadlist>
  <rcr:road gml:id="8620">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#5285" rcr:neighbour="9341"/>
      <gml:directedEdge orientation="+" xlink:href="#6940"/>
      <gml:directedEdge orientation="-" xlink:href="#5655" rcr:neighbour="9515"/>
      <gml:directedEdge orientation="+" xlink:href="#6939"/>
    </gml:Face>
  </rcr:road>
  <rcr:road gml:id="8621">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#6575" rcr:neighbour="9630"/>
      <gml:directedEdge orientation="+" xlink:href="#7372"/>
      <gml:directedEdge orientation="+" xlink:href="#7371" rcr:neighbour="9958"/>
      <gml:directedEdge orientation="+" xlink:href="#7370"/>
    </gml:Face>
  </rcr:road>
  <rcr:road gml:id="8622">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#5126" rcr:neighbour="9716"/>
      <gml:directedEdge orientation="+" xlink:href="#5296"/>
      <gml:directedEdge orientation="+" xlink:href="#5295" rcr:neighbour="9827"/>
      <gml:directedEdge orientation="+" xlink:href="#5294"/>
    </gml:Face>
  </rcr:road>
  ...
  <rcr:road gml:id="9984">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#6149" rcr:neighbour="9351"/>
      <gml:directedEdge orientation="-" xlink:href="#6355" rcr:neighbour="8952"/>
      <gml:directedEdge orientation="-" xlink:href="#4484" rcr:neighbour="8960"/>
    </gml:Face>
  </rcr:road>
</rcr:roadlist>
```

Figure 15. GML Road List

3.3.3. Creating a scenario

The scenario file is also XML formatted file, and contains a list of the entities that compose the simulation initial state, including the starting fires, refuges, civilians, agents, etc. Each element of the xml file has two attributes. The first determines the type of entity being created (fire, refuge, ambulance, ambulance centre, etc.) and the second determines where the location of the entity in the map at the beginning of the simulation. The location is a number that refers to the `id` of an entity in the GML file (either a Building or a Road).

There are two tools that can assist in creating a scenario: Scenario Editor and Random Scenario. Please refer the RoboCup Rescue Simulator Manual for information of how to run these tools.

Figure 16 shows a reduced representation of scenario file created for University of São Paulo map.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<scenario:scenario xmlns:scenario="urn:roborescue:map:scenario">
  <scenario:refuge scenario:location="8509"/>
  <scenario:refuge scenario:location="8600"/>
  <scenario:refuge scenario:location="8567"/>
  <scenario:fire scenario:location="8544"/>
  <scenario:fire scenario:location="8601"/>
  <scenario:fire scenario:location="8612"/>
  <scenario:fire scenario:location="8513"/>
  <scenario:fire scenario:location="8577"/>
  <scenario:firestation scenario:location="8598"/>
  <scenario:firestation scenario:location="8589"/>
  <scenario:firestation scenario:location="8590"/>
  <scenario:ambulancecentre scenario:location="8560"/>
  <scenario:ambulancecentre scenario:location="8553"/>
  <scenario:gasstation scenario:location="8533"/>
  <scenario:gasstation scenario:location="8575"/>
  <scenario:civilian scenario:location="8892"/>
  <scenario:civilian scenario:location="9234"/>
  <scenario:civilian scenario:location="9834"/>
  <scenario:civilian scenario:location="8723"/>
  <scenario:civilian scenario:location="9030"/>
  <scenario:firebrigade scenario:location="8999"/>
  <scenario:firebrigade scenario:location="9090"/>
  <scenario:firebrigade scenario:location="8934"/>
  <scenario:firebrigade scenario:location="9645"/>
  <scenario:policeforce scenario:location="9045"/>
  <scenario:policeforce scenario:location="9239"/>
  <scenario:policeforce scenario:location="9139"/>
  <scenario:ambulanceteam scenario:location="9211"/>
  <scenario:ambulanceteam scenario:location="9639"/>
  <scenario:ambulanceteam scenario:location="9268"/>
  <scenario:ambulanceteam scenario:location="9812"/>
  <scenario:hydrant scenario:location="9011"/>
</scenario:scenario>
```

Figure 16. Scenario Layout

Some types of entities can be located only on Buildings, others only on Roads, and other yet in both. The following list shows the types of entities and where they can be located at.

- Building
  - fire
  - refuge
  - firestation
  - ambulancecentre
  - policeoffice
  - gasstation

    - ambulanceteam
    - policeforce
    - firebrigade
    - civilian
- Road
    - hydrant
    - ambulanceteam
    - policeforce
    - firebrigade
    - civilian