A Standard for Map Data Representation

IEEE 1873-2015 Facilitates Interoperability Between Robots


©ISTOCKPHOTO.COM/BRGFX

By Francesco Amigoni, Wonpil Yu, Torsten Andre, Dirk Holz, Martin Magnusson, Matteo Matteucci, Hyungpil Moon, Masashi Yokotsuka, Geoffrey Biggs, and Raj Madhavan

Digital Object Identifier 10.1109/MRA.2017.2746179
Date of publication: 16 January 2018

The availability of environment maps for autonomous robots enables them to complete several tasks. A new IEEE standard, IEEE 1873-2015, *Robot Map Data Representation for Navigation* (*MDR*) [15], sponsored by the IEEE Robotics and Automation Society (RAS) and approved by the IEEE Standards Association Standards Board in September 2015, defines a common representation for two-dimensional (2-D) robot maps and is intended to facilitate interoperability among navigating robots. The standard defines an extensible markup language (XML) data format for exchanging maps between different systems. This article illustrates how metric maps, topological maps, and their combinations can be represented according to the standard.

Scope of the IEEE 1873-2015 Standard

A map models a robot's knowledge about its environment [3]. For instance, a map represents the positions of the obstacles that could limit its movements. Map availability is thus a requirement for the autonomous execution of several tasks by a robot. This article overviews the main ideas of the IEEE 1873-2015 MDR [15] and presents some illustrative examples of its use. The standard focuses on interoperability and provides specifications for representing 2-D metric and topological maps to ease the process of exchanging map data among robots, computers, and other devices. It focuses on maps used for the navigation of mobile robots, which usually include representations of free space and obstacles. IEEE 1873-2015 1) defines some elements related to 2-D robot maps for navigation and a data model for each element and 2) defines an XML data format for map data exchange.

Because the standard aims at defining a common representation for exchanging maps between devices, the data format it specifies provides human-readable textual information for easy inspection, validation, and debugging, even though it is not optimized for map processing (e.g., map building). The general setting considered by the standard is that of robot systems using their own (different) internal map representations that are translated to and from the standard data format only when robots need to exchange map data.

The standard can find a natural application in different situations. For instance, the ability to unambiguously share robot map data, including uncertainties, is useful when robots are collaboratively constructing the map of an environment. The standard can also promote the development of good experimental methodologies for mobile robotics by providing a common format for distributing data sets and for easing the comparison and evaluation of maps obtained with different systems. Maps encoded in a standard format will also make it possible to perform comparisons and evaluations for a long time when compared to maps encoded in some proprietary data format. More generally, the standard can enable interoperability between systems developed by different manufacturers that operate in different environments.

There are several standardization activities carried out in robotics. These activities aim at defining aspects of the performance of robot systems to set safety requirements, to ease interoperability, and to support scientific research and industrial development [7]. Current standardization activities of robot technologies are mostly related to the safety of industrial robots, such as those performed within the International Organization for Standardization (ISO) [13], the Robotic Industries Association, and the American National Standards Institute. The ontology standard IEEE 1872-2015 is the first one sponsored by RAS, and it addresses interoperability by defining a formal reference vocabulary for communicating knowledge about robotics and automation, both among robots and between robots and humans [16]. Other standards for interoperability in robot software development [21] and in robot operations have recently been proposed. Individual countries have also put effort into standardization activities in robotics. Japan, for instance, produced a vocabulary standard on robots and robotic devices [22], while Korea has published more than 150 standards documents in industrial and service robotics [23].

Developed specifically for robotic navigation, IEEE 1873-2015 considers, among others, issues related to the uncertainty of map elements poses. In this sense, it differs from other standards for map specification already published or under development from other standard developing organizations, including the ISO TC 211 Geography Markup Language (GML) [17], the Open Geospatial Consortium Web Map Service (WMS) [18], and the City GML [19]. Another distinctive characteristic of the standard is that it can represent arbitrary combinations of metric and topological maps.

Overview of the IEEE 1873-2015 Standard

The IEEE 1873-2015 standard defines the concept of the global map, which is a collection of local maps. A local map is either a metric or a topological map. In the context of the standard, metric maps include grid and geometric maps. A grid map decomposes the representation of an environment into (equal) square cells that constitute atomic pieces of information [4], [8]. Typically, a cell describes whether (or the probability of) the corresponding part of the environment belongs to the free space or to an obstacle. A geometric map is composed of a list of continuous geometric features (such as points and line segments) [1], [6]. A topological map represents an environment in the form of a graph consisting of a set of nodes and the edges connecting them (see, e.g., [2], [10], and [12]). Topological map nodes represent characteristic features of a part of an environment (e.g., poses, sensor readings, unique signatures of the perceived data, or metric maps). Edges of a topological map represent directed (one-hop) connections between neighboring nodes. In metric maps, the distance between any two elements can be computed, while, for topological maps, this can apply to only some elements.

The standard defines these elements at two levels of abstraction. At first, it defines the data models, then it defines the actual data formats corresponding to the data models. Here, we report only some aspects of the standard to give the flavor of the approach followed and to understand how it can be used. See the full document for full details [15]. The data model is defined according to the unified modeling language (UML) [11]. For instance, Figure 1 shows a fragment of the representation of the data model relative to local maps. Grid maps, geometric maps, and topological maps are inherited (see arrows) from `LocalMap` and are further defined in the standard. Attributes are reported as *`name: type`*, where the type can be primitive (like `string` or `double`) or can be further specified by the standard (like `LocalMapTransform` and `LocalMapType`). Optional attributes have a cardinality ranging from 0 to 1 (and are identified by `[0..1]` in Figure 1), while mandatory attributes have the cardinality 1 (all the others in Figure 1). A local map is associated to a (right-handed) Cartesian coordinate system, whose pose can be specified either absolutely (i.e., it is geolocalized) or relatively to another coordinate system [i.e., an offset (x, y, θ) indicates a rigid transformation between the two coordinate systems], as detailed in the "Combined Representations of

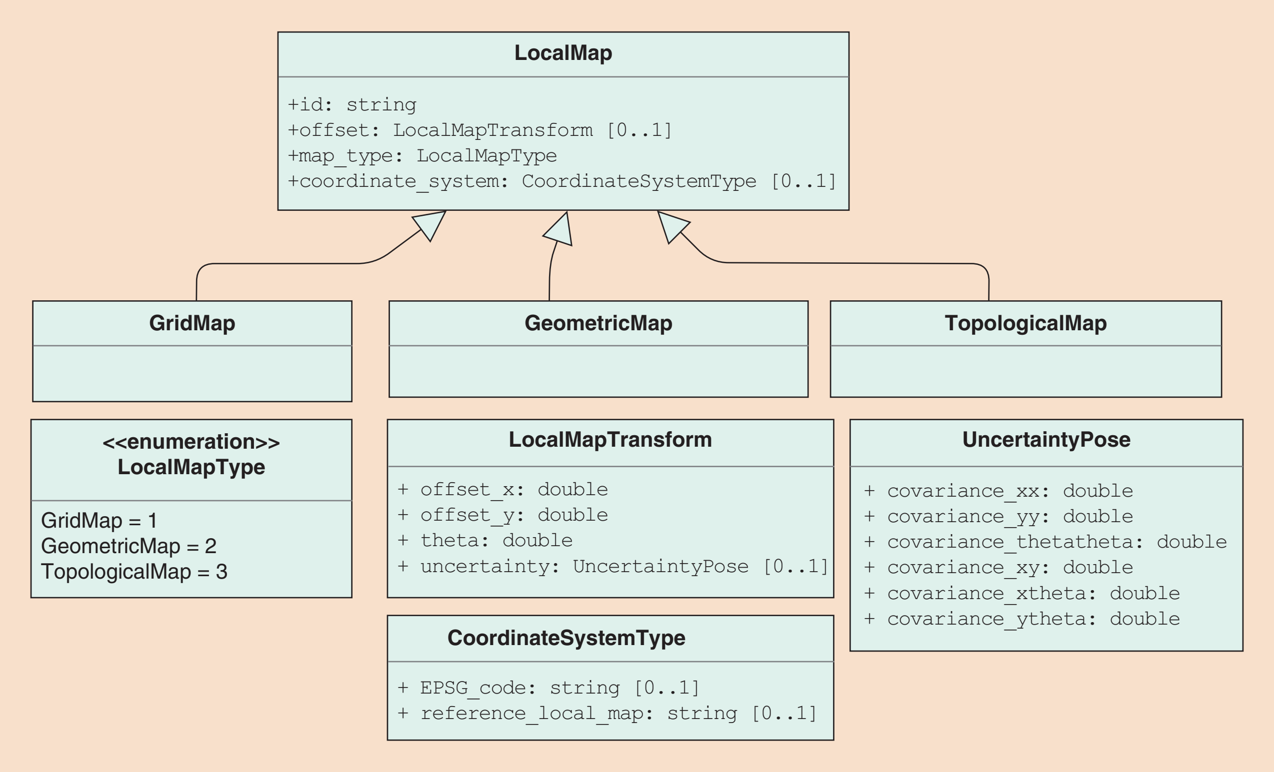

Figure 1. The UML data model of a local map.

Different Local Maps" section. The data model of Figure 1 includes the uncertainty about the relative pose of a local map, which is expressed as a 3 × 3 covariance matrix. The standard also defines UML data models for grid, geometric, and topological maps (whose details are not shown here).

The UML data models are then associated to data formats described in XML. Using XML as an exchange data format provides a number of advantages. First, XML is a platform-independent language, and all major programming languages and operating systems include XML parsers. Second, it is well known by many professionals and data can be stored in a human-readable format, allowing easy understanding and debugging of map files. Finally, XML allows automatic validation and checking of exchanged data, using an XML schema definition (XSD). Drawbacks include the commonly criticized large overhead of the language. The standard defines the data format as an XSD. Code 1 reports the data format for the elements of the data model of Figure 1, as defined in the standard. The XSD type `LocalMap` includes the attribute `mdr_version` (line 9) that stores the minimum version of the IEEE 1873-2015 standard according to which the `LocalMap` can be validated. Also, `LocalMap` optionally stores some metadata (line 3) that can include author names and e-mails, the license under which the map is distributed, copyright owners, a textual description (e.g., the robot platforms used to create the map and the map location), and the date and time of creation and last modification of the map. `LocalMaps` can be of types `GridMap`, `TopologicalMap`, or `GeometricMap` (lines 8 and 21–27). The XSDs for these maps (not shown here) specify the standard format for expressing information like cells for grid maps and points for geometric maps. `CoordinateSystemType` (lines 29–32) specifies if the coordinate system of the local map is georeferenced or if its pose is expressed relatively to that of another local map. In practice, the meaning of `LocalMapTransform` depends on `CoordinateSystemType`, as illustrated in the "Combined Representations of Different Local Maps" section. `MapArray` represents the collection of `LocalMaps` composing a global map (lines 43–49). The XSD data format defined by the standard is eventually instantiated in XML files describing specific global maps.

The IEEE 1873-2015 Standard in Practice

In this section, we present some illustrative examples of employment of the standard to represent grid maps, geometric maps, topological maps, and their combinations. Note that the reported use cases are not meant to be exhaustive of the possibilities offered by the standard.

Representing Grid Maps

Consider the simple (synthetic) environment of Figure 2, which is a small 2 m × 2 m room with an L-shaped obstacle in the middle and which is represented by a grid map describing the occupancy of the environment (white and dark cells in Figure 2). Code 2 shows the file representing this grid map in the XML data format defined by the standard. After some initial information about the encoding of the XML file and the location of the XSD file that can be used for validating the

```
<xs:complexType name=”LocalMap” abstract=”true”>
  <xs:sequence>
    <xs:element name=”metadata” type=”Metadata”/>
    <xs:element name=”offset” type=”LocalMapTransform” minOccurs=”0” maxOccurs=”1”/>
    <xs:element name=”coordinate_system” type=”CoordinateSystemType” minOccurs=”0”
        maxOccurs=”1”/>
  </xs:sequence>
  <xs:attribute name=”id” type=”xs:string” use=”required”/>
  <xs:attribute name=”map_type” type=”LocalMapType” use=”required”/>
  <xs:attribute name=”mdr_version” type=”xs:string” use=”required”/>
</xs:complexType>
<!- ========================================================================= ->
<xs:complexType name=”LocalMapTransform”>
  <xs:sequence>
    <xs:element name=”uncertainty” minOccurs=”0” maxOccurs=”1” type=”UncertaintyPose”/>
  </xs:sequence>
  <xs:attribute name=”offset_x” type=”xs:double” use=”required”/>
  <xs:attribute name=”offset_y” type=”xs:double” use=”required”/>
  <xs:attribute name=”theta” type=”xs:double” use=”required”/>
</xs:complexType>
<!- ========================================================================= ->
<xs:simpleType name=”LocalMapType”>
  <xs:restriction base=”xs:integer”>
    <xs:enumeration value=”1”/> <!- The number ‘1’ means “GridMap”. ->
    <xs:enumeration value=”2”/> <!- The number ‘2’ means “GeometricMap”. ->
    <xs:enumeration value=”3”/> <!- The number ‘3’ means “TopologicalMap”. ->
  </xs:restriction>
</xs:simpleType>
<!- ========================================================================= ->
<xs:complexType name=”CoordinateSystemType”>
  <xs:attribute name=”EPSG_code” type=”xs:string” use=”optional”/>
  <xs:attribute name=”reference_local_map” type=”xs:string” use=”optional”/>
</xs:complexType>
<!- ========================================================================= ->
<xs:complexType name=”UncertaintyPose”>
  <xs:attribute name=”covariance_xx” type=”xs:double” use=”required”/>
  <xs:attribute name=”covariance_yy” type=”xs:double” use=”required”/>
  <xs:attribute name=”covariance_thetatheta” type=”xs:double” use=”required”/>
  <xs:attribute name=”covariance_xy” type=”xs:double” use=”required”/>
  <xs:attribute name=”covariance_xtheta” type=”xs:double” use=”required”/>
  <xs:attribute name=”covariance_ytheta” type=”xs:double” use=”required”/>
</xs:complexType>
<!- ========================================================================= ->
<xs:complexType name=”MapArray”>
  <xs:choice minOccurs=”0” maxOccurs=”unbounded”>
    <xs:element name=”grid_map” type=”GridMap” minOccurs=”0” maxOccurs=”unbounded”/>
    <xs:element name=”geometric_map” type=”GeometricMap” minOccurs=”0”
        maxOccurs=”unbounded”/>
    <xs:element name=”topological_map” type=”TopologicalMap” maxOccurs=”unbounded”
        minOccurs=”0”/>
  </xs:choice>
</xs:complexType>
```

Code 1. A fragment of the XSD data format of a local map representing the data model of Figure 1. At the bottom, a `MapArray` is defined as a collection of local maps (grid, geometric, or topological). At the top, a `LocalMap` is defined according to attributes that are in turn defined (`LocalMapTransform`, `LocalMapType`, `CoordinateSystemType`, and `UncertaintyPose`).

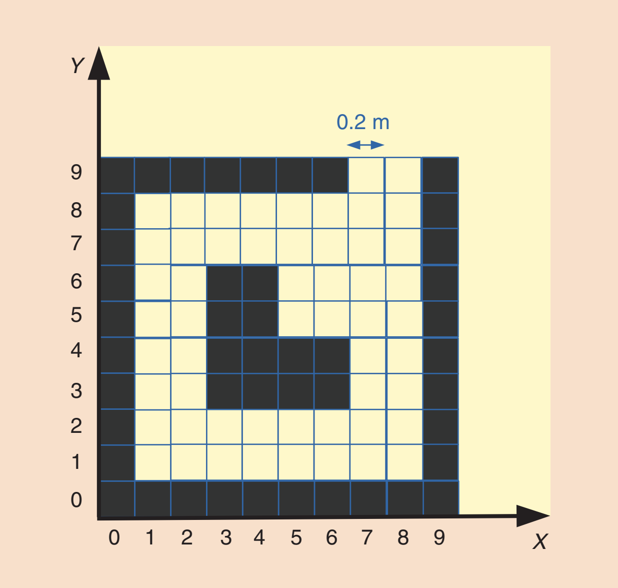

Figure 2. An example environment represented as a grid map (the light cells are free space and the dark cells are obstacles). The local coordinate system is shown.

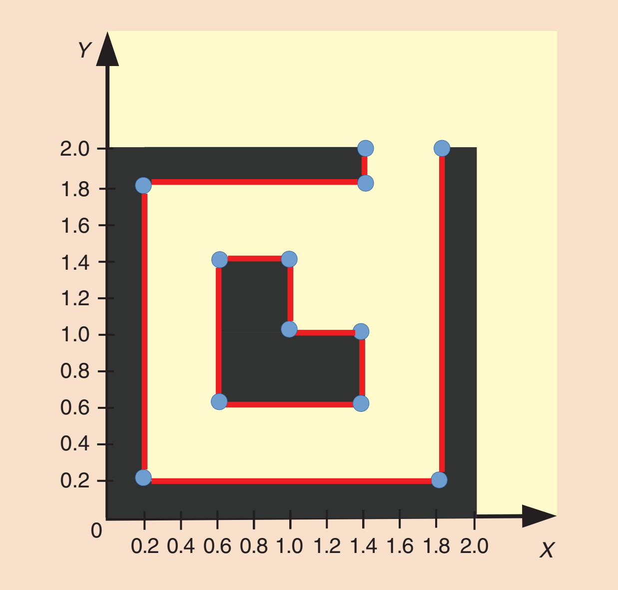

Figure 3. The example environment represented as a geometric map (points are in blue and line segments are in red). The local coordinate system is shown (the unit is meters).

content (lines 1–5), the grid map representation is introduced (line 7) specifying its identification (id) (`GridMap`, in this case), its type (1 means grid map, 2 means geometric map, and 3 means topological map; see Figure 1 and Code 1, lines 21–27), the total number of cells along the *x* and the *y* axis of the map (i.e., the coordinates of cells are in the ranges 0 to `num_cells_x -1` and 0 to `num_cells_y -1`), and the length of an edge of a cell expressed in meters (`resolution`). Then, metadata can be optionally reported (for simplicity, metadata are omitted in Code 2, line 9).

The coordinate system is not specified with respect to any other coordinate system, but the (0, 0, 0) `offset` is (lines 11–14). This means that the absolute position of this grid map is unknown, i.e., that its local coordinate system is placed in the (0, 0, 0) pose of an arbitrarily placed global coordinate system. In this example, uncertainty about the pose of the coordinate system is zero (line 13).

Then, the palette is defined (lines 16–18), which specifies the values that can be assigned to each cell and their meaning (in natural language). In the example, the values range from zero to 255, which correspond to a probability of zero and to a probability of 1 of being occupied, respectively. Intermediate values correspond to intermediate probabilities, e.g., value 100 corresponds to the probability $100/255 = 0.39$ of being occupied. Different palettes can be defined by the user, e.g., another palette could specify that all values from zero to 127 indicate a free cell, and all those from 128 to 255 mean an occupied cell. A user can also define a special default value in the palette (e.g., –1) that is given to cells for which no significant value is present. A value for each cell of the map must be specified.

Finally, all the cells composing the grid map are listed (lines 20–36). Cells are represented in blocks, following an idea similar to that of quad trees. For example, `<cell x="0" y="0" width="1" height="10" value="255"/>` (line 22) represents the leftmost vertical wall that starts with the cell at coordinates (0, 0), has a width (along the *x* axis of the local coordinate system) of 1 and height (along the *y* axis) of 10, and all of its ten cells have the value 255 (i.e., represent an obstacle). The same happens for cells representing the other walls and for those representing the obstacle. Free cells are represented similarly, but with the value zero. In the current version of the standard, blocks of cells can only be rectangular with size `width` × `height` and shall not overlap.

The possibility of representing blocks of cells with the same value saves memory compared to representing each cell as an individual entity (a single cell, e.g., the bottom cell of the leftmost vertical wall, can be represented as `<cell x="0" y="0" value="255"/>`). We considered some grid maps taken from public repositories and compared the size of the files representing them when using the message type `nav_msgs/OccupancyGrid` of the Robot Operating System (ROS) navigation stack [24] and the XML format defined by the standard. Despite their human readability, the files of the standard representation have an overall size comparable with that of their ROS counterparts and, in some cases, smaller.

Representing Geometric Maps

We consider the same environment described in the previous section but with it now modeled by geometric primitives (points and line segments), as shown in Figure 3. Code 3 shows the content of the file with the XML standard representation of the map. Also, in this case, after some initial information (lines 1–5), the geometric map is introduced (line 7) by specifying its id (`GeometricMap`) and its type (2 means geometric map). As in the case of the grid map, the local

```
1 <?xml version=”1.0” encoding=”UTF-8”?>
2 <mdr:maps
3     xmlns:mdr=”http://www.example.org/mdr”
4     xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
5     xsi:schemaLocation=”http://www.example.org/mdr␣MDR_schema.xsd”>
6
7 <grid_map id=”GridMap” maptype=”1” num_cells_x=”10” num_cells_y=”10”
        resolution=”0.2”>
8
9 <metadata> <!--…omitted…--> </metadata>
10
11 <coordinate_system/>
12 <offset offset_x=”0.0” offset_y=”0.0” theta=”0.0”>
13   <uncertainty covariance_xx=”0.0” covariance_yy=”0.0” covariance_thetathe-
        ta=”0.0” covariance_xy=”0.0” covariance_xtheta=”0.0” covariance_ythe-
        ta=”0.0”/>
14 </offset>
15
16 <palette_elements>
17   <palette value_start=”0” value_end=”255” meaning=”A␣value␣0...255␣corresponds␣
        to␣an␣occupation␣probability␣0.0...1.0.”/>
18 </palette_elements>
19
20 <cells>
21 <!- Wall Cells ->
22 <cell x=”0” y=”0” width=”1” height=”10” value=”255”/>
23 <cell x=”1” y=”0” width=”8” height=”1” value=”255”/>
24 <cell x=”9” y=”0” width=”1” height=”10” value=”255”/>
25 <cell x=”1” y=”9” width=”6” height=”1” value=”255”/>
26 <!- Obstacle Cells ->
27 <cell x=”3” y=”3” width=”2” height=”4” value=”255”/>
28 <cell x=”5” y=”3” width=”2” height=”2” value=”255”/>
29 <!- Free Space Cells ->
30 <cell x=”1” y=”1” width=”2” height=”8” value=”0”/>
31 <cell x=”3” y=”1” width=”4” height=”2” value=”0”/>
32 <cell x=”7” y=”1” width=”2” height=”4” value=”0”/>
33 <cell x=”3” y=”7” width=”2” height=”2” value=”0”/>
34 <cell x=”5” y=”5” width=”4” height=”4” value=”0”/>
35 <cell x=”7” y=”9” width=”2” height=”1” value=”0”/>
36 </cells>
37
38 </grid_map>
39
40 </mdr:maps>
```

Code 2. The file representing the grid map of Figure 2. After the initial information and the metadata, the coordinate system of the map is specified, the meaning of the values of the cells is defined, and the actual cells composing the map are listed (see the text for further details, and the file is available as supplementary material accompanying this article in IEEE *Xplore*).

coordinate system is aligned to an arbitrarily placed global coordinate system (lines 11–14).

The rest of the XML representation of `GeometricMap` lists the primitive geometric elements, namely, points (lines 17–29) and line segments (lines 31–52), that compose the map. A point is naturally represented by its two coordinates x and y expressed in the local coordinate system. The uncertainty about the position of a point, expressed as a 2×2 covariance matrix, can be reported. For instance, in line 19, `<uncertainty covariance_xx=”0.1” covariance_xy=”0.0” covariance_yy=”0.1”/>` means that, for the corresponding point, the variance of x is 0.1, that of y is 0.1, and the covariance between x and y is 0. A line segment is represented using the parametrization of Figure 4, which is used in [9] and provides a more realistic representation of uncertainty, expressed as covariance on parameters, when compared to the naive parametrization based on the coordinates of the endpoints. The uncertainty on the position

```
<?xml version=″1.0″ encoding=″UTF-8″?>
<mdr:maps
    xmlns:mdr=″http://www.example.org/mdr″
    xmlns:xsi=″http://www.w3.org/2001/XMLSchema-instance″
    xsi:schemaLocation=″http://www.example.org/mdr␣MDR_schema.xsd″>

<geometric_map id=″GeometricMap″ maptype=″2″>

<metadata> <!-…omitted… -> </metadata>

<coordinate_system/>
<offset offset_x=″0.0″ offset_y=″0.0″ theta=″0.0″>
  <uncertainty covariance_xx=″0.0″ covariance_yy=″0.0″ covariance_thetatheta=″0.0″ covariance_xy=″0.0″ covariance_xtheta=″0.0″ covariance_ytheta=″0.0″/>
</offset>

<elements>
<!- Points Representing Wall Corners ->
  <point x=″0.2″ y=″0.2″>
    <uncertainty covariance_xx=″0.1″ covariance_xy=″0.0″ covariance_yy=″0.1″/>
  </point>
<point x=″0.2″ y=″1.8″>
  <uncertainty covariance_xx=″0.1″ covariance_xy=″0.0″ covariance_yy=″0.1″/>
</point>
  <point x=″1.4″ y=″1.8″>
    <uncertainty covariance_xx=″0.1″ covariance_xy=″0.0″ covariance_yy=″0.1″/>
</point>
<!-…omitted… ->
<!- Points Representing Obstacle Corners ->
<!-…omitted… ->

<!- Line Segments Representing Wall Edges ->
  <line_segment rho=″0.2″ alpha=″1.5708″ psi_a=″0.2″ psi_b=″1.8″>
    <uncertainty covariance_rhorho=″0.1″ covariance_rhoalpha=″0.0″ covariance_rhopsi_a=″0.0″
covariance_rhopsi_b=″0.0″ covariance_alphaalpha=″0.1″ covariance_alphapsi_a=″0.0″
covariance_alphapsi_b=″0.0″ covariance_psi_apsi_a=″0.1″ covariance_psi_apsi_b=″0.0″
covariance_psi_bpsi_b=″0.1″/>
  </line_segment>
  <line_segment rho=″0.2″ alpha=″0.0″ psi_a=″0.2″ psi_b=″1.8″>
    <uncertainty covariance_rhorho=″0.1″ covariance_rhoalpha=″0.0″ covariance_rhopsi_a=″0.0″
covariance_rhopsi_b=″0.0″ covariance_alphaalpha=″0.1″ covariance_alphapsi_a=″0.0″
covariance_alphapsi_b=″0.0″ covariance_psi_apsi_a=″0.1″ covariance_psi_apsi_b=″0.0″
covariance_psi_bpsi_b=″0.1″/>
  </line_segment>
  <line_segment rho=″1.8″ alpha=″1.5708″ psi_a=″0.2″ psi_b=″1.4″>
    <uncertainty covariance_rhorho=″0.1″ covariance_rhoalpha=″0.0″ covariance_rhopsi_a=″0.0″
covariance_rhopsi_b=″0.0″ covariance_alphaalpha=″0.1″ covariance_alphapsi_a=″0.0″
covariance_alphapsi_b=″0.0″ covariance_psi_apsi_a=″0.1″ covariance_psi_apsi_b=″0.0″
```

Code 3. The file representing the geometric map of Figure 3. After the initial information and the metadata, the coordinate system of the map is specified, and the points and line segments composing the map are listed (see the text for further details, and the complete file is available as supplementary material accompanying this article in IEEE *Xplore*). (Continued on the next page.)

```
  covariance_psi_bpsi_b=”0.1”/>
  </line_segment>
<!-…omitted… ->
<!- Line Segments Representing Obstacle Edges ->
<!-…omitted… ->
</elements>

</geometric_map>

</mdr:maps>
```

Code 3. (continued) The file representing the geometric map of Figure 3. After the initial information and the metadata, the coordinate system of the map is specified, and the points and line segments composing the map are listed (see the text for further details, and the complete file is available as supplementary material accompanying this article in IEEE *Xplore*).

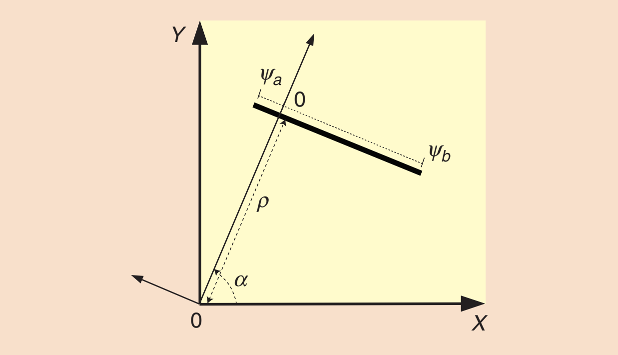

Figure 4. A line segment representation. In the XML data format, ρ is **`rho`** in the standard XML format, α is **`alpha`**, ψ_a is **`psi_a`**, and ψ_b is **`psi_b`**.

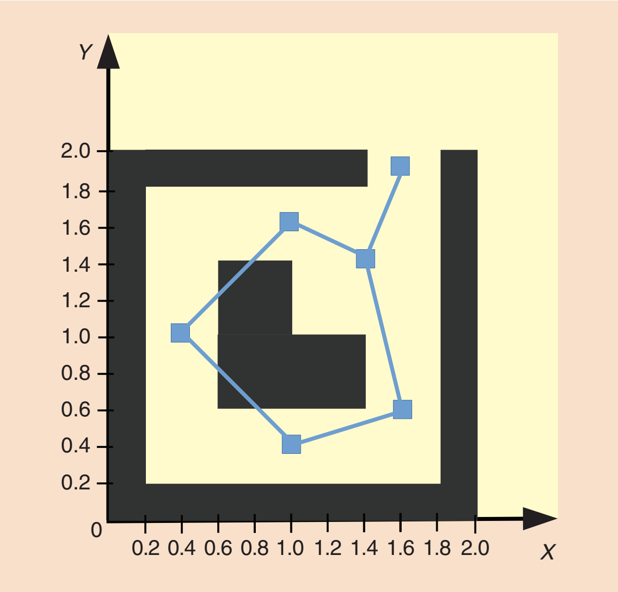

Figure 5. The example environment represented as a topological map (nodes are the light blue squares and edges are the light blue line segments). The local coordinate system is shown (the unit is meters).

of a line segment is expressed by listing the elements of the covariance matrix relative to the four parameters.

Note that the geometric representation presented is just for illustrative purposes and is not optimized. It is redundant because the points representing the corners of the walls and of the obstacle could be derived as the points at which the line segments touch each other (or, more realistically, at which the distance between the line segments is less than a threshold). It is possible to represent geometric maps composed only of points or only of line segments.

We compared the size of the files containing scans acquired by laser range scanners (which are point-only geometric maps) represented either in the Player format [20] or in the standard format. Results show that the size of the files represented with the standard is approximately three times larger than that of the Player's files due to the overhead introduced by the XML representation. Note that the current version of the standard does not prescribe any way (similar to the blocks of cells for grid maps) to limit the size of the geometric maps.

Representing Topological Maps

Consider a topological map of the same environment as in the previous sections, as shown in Figure 5. The topological map models the environment as a graph with some places (which correspond to nodes) and their connections (which correspond to edges). Code 4 shows the file representing the topological map in the XML standard format. The id is `TopologicalMap`, `maptype 3` refers to topological maps (line 7), and its coordinate system is aligned to an arbitrarily placed global coordinate system (lines 11–14).

In the standard, each node is described by a symbolic id and, optionally, by its location and its properties (lines 16–32). In our example, we specify the locations of all nodes (as coordinates x and y in the coordinate system of `TopologicalMap`). Custom properties of a node can be added by the user as a list. Typical examples of properties are the sensor data collected at the node location and the features extracted from them. The algorithm shows an example of a property associated to `node5` (lines 22–31), which represents the distance between the location of the node and the nearest obstacle in the environment (lines 24–29). Note that the number of properties associated to a node is represented explicitly by `property_num`.

Finally, (oriented) edges are specified by their ids, their tail and head nodes (starting and ending nodes, respectively), and, optionally, their properties (lines 34–56). Our example

```
<?xml version=”1.0” encoding=”UTF-8”?>
<mdr:maps
   xmlns:mdr=”http://www.example.org/mdr”
   xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
   xsi:schemaLocation=”http://www.example.org/mdr␣MDR_schema.xsd”>

<topological_map id=”TopologicalMap” maptype=”3”>

<metadata> <!-…omitted… -> </metadata>

<coordinate_system/>
<offset offset_x=”0.0” offset_y=”0.0” theta=”0.0”>
  <uncertainty covariance_xx=”0.0” covariance_yy=”0.0” covariance_thetatheta=”0.0” covariance_xy=”0.0” covariance_xtheta=”0.0” covariance_ytheta=”0.0”/>
</offset>

<nodes>
  <node id=”node0” property_num=”0”><location x=”1.6” y=”1.9”/></node>
  <node id=”node1” property_num=”0”><location x=”1.4” y=”1.4”/></node>
  <node id=”node2” property_num=”0”><location x=”1.0” y=”1.6”/></node>
  <node id=”node3” property_num=”0”><location x=”0.4” y=”1.0”/></node>
  <node id=”node4” property_num=”0”><location x=”1.0” y=”0.4”/></node>
  <node id=”node5” property_num=”1”><location x=”1.6” y=”0.6”/>
    <properties>
      <property>
        <name>DistNearest</name>
        <value>0.2</value>
        <typename>float</typename>
        <description>Distance to the nearest obstacle in meter unit</description>
      </property>
    </properties>
  </node>
</nodes>

<edges>
  <edge id=”edge0” property_num=”0” head_node=”node0” tail_node=”node1”></edge>
  <edge id=”edge1” property_num=”0” head_node=”node1” tail_node=”node2”></edge>
  <edge id=”edge2” property_num=”0” head_node=”node2” tail_node=”node3”></edge>
  <edge id=”edge3” property_num=”0” head_node=”node3” tail_node=”node4”></edge>
  <edge id=”edge4” property_num=”0” head_node=”node4” tail_node=”node5”></edge>
  <edge id=”edge5” property_num=”2” head_node=”node5” tail_node=”node1”
    <properties>
      <property>
        <name>EdgeLength</name>
        <value>0.6325</value>
        <typename>float</typename>
        <description>Edge length in meter unit</description>
      </property>
      <property>
        <name>EdgeTerrain</name>
        <value>flat</value>
        <typename>string</typename>
        <description>Type of terrain enconuntered when traveling along the edge</description>
      </property>
    </properties>
```

Code 4. The file representing the topological map of Figure 5. After the initial information and the metadata, the coordinate system of the map is specified, and the nodes and edges composing the map are listed (see the text for further details, and the complete file is available as supplementary material accompanying this article in IEEE *Xplore*). (Continued on the next page.)

```
55   </edge>
56  </edges>
57
58 </topological_map>
59
60 </mdr:maps>
```

Code 4. (continued) The file representing the topological map of Figure 5. After the initial information and the metadata, the coordinate system of the map is specified, and the nodes and edges composing the map are listed (see the text for further details, and the complete file is available as supplevmentary material accompanying this article in IEEE *Xplore*).

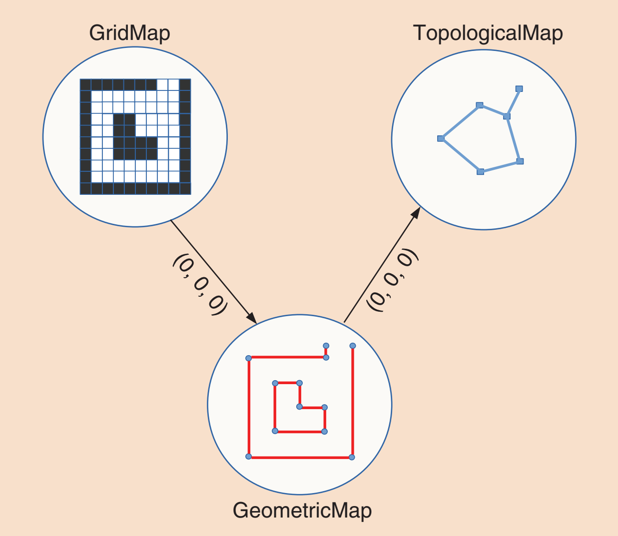

Figure 6. The graph defined by the relationships between the coordinate systems of the local maps of Code 5.

specifies two properties for `edge5` (lines 40–55), namely, its length (lines 42–47) and the type of terrain (e.g., flat) that a robot expects to negotiate when traveling along the connection represented by that edge (lines 48–53).

Combined Representations of Different Local Maps

One of the most interesting features offered by the IEEE 1873-2015 standard is the possibility of representing, in a coherent way, a given environment using a combination of grid, geometric, and topological local maps. The collection of these maps is a global map that is stored in a single XML file.

For instance, the room with an obstacle in the middle that we have used as a running example in the previous sections can be represented by a global map that combines the grid map of Code 2, the geometric map of Code 3, and the topological map of Code 4. Code 5 shows the structure and some of the content of the XML file representing this global map.

In the global map, the coordinate system of the grid map (lines 11–14) is aligned to that of an arbitrarily placed global coordinate system (as in the "Representing Grid Maps" section). The pose of the coordinate system of the geometric map is specified relative to that of the grid map (line 25) and, being the offset (0, 0, 0) (lines 26–28), the two coordinate systems are aligned. The uncertainty of the relative pose of `GeometricMap` with respect to `GridMap` is zero. Finally, the coordinate system of the topological map is aligned with that of `GeometricMap`, and there is no uncertainty as to its relative pose (lines 37–41). The relationships between the coordinate systems of the three local maps define the graph of Figure 6, whose nodes are local maps and edges are the relative poses (offsets) between their coordinate systems. The graph shows what we have just discussed, i.e., that the coordinate system of `GeometricMap` is aligned to `GridMap` and that, similarly, the coordinate system of `TopologicalMap` is aligned to `GeometricMap`.

In the standard, the pose of a local map can be specified as follows:

- *Null*: When `offset` is not specified, meaning that the coordinate system of the local map is at an unknown pose. This is the case, e.g., for purely topological maps.
- *Absolute*: When `offset` is specified together with a European Petroleum Survey Group (EPSG) code [25] that defines a geolocalized pose. An EPSG code uniquely identifies a pose around the world using some geodetic parameters (like latitude and longitude). For instance, the EPSG code 2000 refers to a pose at Anguilla, in the Caribbean Sea. When the offset of a local map is specified according to this absolute pose, the local map is then geolocalized.
- *Relative to the coordinate system of another local map*: When `offset` is specified together with the id of a local map whose coordinate system acts as a reference according to which the relative pose of the local map is calculated. See, e.g., the cases of `GeometricMap` and `TopologicalMap` of Code 5.
- *Relative to an arbitrarily placed coordinate system* (i.e., relative to a coordinate system whose pose in unknown): When `offset` is specified, but neither an EPSG code nor reference local map is defined. This is the case, for example, of `GridMap` of Code 5.

The standard constrains the mutual references between local maps of the same global map to avoid inconsistencies between their relative poses (e.g., circular references in which the relative pose of a local map is ultimately expressed with respect to the pose of the local map itself).

Global maps can contain an arbitrary number of local maps of any type. For instance, a global map can be used to represent grid maps independently built by different robots before they are merged. In this case, the relationships between the coordinate systems of the local maps can be known (e.g., when the relative poses of the robots are known) or unknown (e.g., when the relative poses of the robots are unknown). Another example of global maps that can contain several local maps of the same type is that of data sets. In this case, the local maps

```
<?xml version=”1.0” encoding=”UTF-8”?>
<mdr:maps
     xmlns:mdr=”http://www.example.org/mdr”
     xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
     xsi:schemaLocation=”http://www.example.org/mdr␣MDR_schema.xsd”>

<grid_map id=”GridMap” maptype=”1” num_cells_x=”10” num_cells_y=”10” resolution=”0.2”>

<metadata> <!-…omitted… -> </metadata>

<coordinate_system/>
<offset offset_x=”0.0” offset_y=”0.0” theta=”0.0”>
  <uncertainty covariance_xx=”0.0” covariance_yy=”0.0” covariance_thetatheta=”0.0” covariance_xy=”0.0” covariance_xtheta=”0.0” covariance_ytheta=”0.0”/>
</offset>

<!-…elements of the grid map… ->

</grid_map>


<geometric_map id=”GeometricMap” maptype=”2”>

<metadata> <!-…omitted… -> </metadata>

<coordinate_system reference_local_map=”GridMap”/>
<offset offset_x=”0.0” offset_y=”0.0” theta=”0.0”>
    <uncertainty covariance_xx=”0.0” covariance_yy=”0.0” covariance_thetatheta=”0.0” covariance_xy=”0.0” covariance_xtheta=”0.0” covariance_ytheta=”0.0”/>
</offset>

<!-…elements of the geometric map… ->

</geometric_map>


<topological_map id=”TopologicalMap” maptype=”3”>

<coordinate_system reference_local_map=”GeometricMap”/>
<offset offset_x=”0.0” offset_y=”0.0” theta=”0.0”>
    <uncertainty covariance_xx=”0.0” covariance_yy=”0.0” covariance_thetatheta=”0.0”
covariance_xy=”0.0” covariance_xtheta=”0.0” covariance_ytheta=”0.0”/>
</offset>

<!-…elements of the topological map… ->

</topological_map>

</mdr:maps>
```

Code 5. The file representing the global map composed of the local maps of Codes 2, 3, and 4. After the initial information, the grid map, the geometric map, and the topological map are specified (see the text for further details, and the complete file is available as supplementary material accompanying this article in IEEE *Xplore*).

can be point (geometric) maps representing the single scans acquired in an environment with a laser range scanner. Multilayer maps [14] and topometric maps [12] can be represented as global maps composed of different local maps. An example of a global map composed of a grid map and of a topological map built by multiple robots and represented using the IEEE 1873-2015 standard is reported in [5]. Topological maps that provide a coarse-grained model of wide areas and metric maps that provide a fine-grained model of small areas of interest could be employed to represent outdoor environments.

Conclusions

This article has presented the main ideas and some application examples of the IEEE 1873-2015, *Robot Map Data Representation for Navigation* standard, which offers a common way to represent maps to be exchanged to facilitate interoperability between robot systems. In particular, we briefly discussed the data model and the data format defined by the standard, and we illustrated simple examples of its applications to the representation of grid, geometric, and topological maps and their compositions.

Future developments will address the possibility of including in the standard more detailed information (e.g., the intensity of points in geometric maps and semantic knowledge in topological maps), of defining more compact map data formats, and of extending the representation to three-dimensional maps.

References

[1] N. Ayache and O. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Autom.*, vol. 5, no. 6, pp. 804–819, 1989.

[2] J. Choi, M. Choi, S. Y. Nam, W. K. Chung, "Autonomous topological modeling of a home environment and topological localization using a sonar grid map," *Autonomous Robots*, vol. 30, no. 4, pp. 351–368, 2011.

[3] H. Christensen and G. Hager, "Sensing and estimation," in *Handbook of Robotics,* B. Siciliano and O. Khatib, Eds. New York: Springer-Verlag, 2008, pp. 87–107.

[4] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Trans. Robot. Autom.*, vol. 3, no. 3, pp. 249–265, 1987.

[5] H.-S. Jin, W. Yu, and H. Moon, "Merging of topological map and grid map using standardized map data representation," *J. Korea Robot. Soc.*, vol. 9, no. 2, pp. 104–110, 2014.

[6] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 376–382, 1991.

[7] R. Madhavan, R. Lakaemper, and T. Kalmar-Nagy, "Benchmarking and standardization of intelligent robotic wystems," in *Proc. Int. Conf. Advanced Robotics (ICAR)*, 2009, pp. 1–7.

[8] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, 1988.

[9] S. Pfister, "Algorithms for mobile robot localization and mapping, incorporating detailed noise modeling and multi-scale feature extraction," Ph.D. dissertation, California Institute of Technology, Pasadena, 2006.

[10] A. Pronobis, "Semantic mapping with mobile robots," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, 2011.

[11] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed. New York: Pearson, 2004.

[12] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intel.*, vol. 99, no. 1, pp. 21–71, 1998.

[13] G. Virk and S. Cameron, "ISO-IEC standardization efforts in robotics," in *Proc. Workshop on Standardized Knowledge Representation and Ontologies for Robotics and Automation*, 2014, pp. 5–6.

[14] H. Zender, O. Martinez Mozos, P. Jensfelt, G. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robot. Auton. Syst.*, vol. 56, no. 6, pp. 493–502, 2008.

[15] IEEE Standards Association. (2015). IEEE standard for robot map data representation for navigation. IEEE *Xplore*. [Online]. Available: https://standards.ieee.org/findstds/standard/1873-2015.html

[16] IEEE Standards Association. (2015). IEEE standard ontologies for robotics and automation. IEEE *Xplore*. [Online]. Available: https://standards.ieee.org/findstds/standard/1872-2015.html

[17] OGC. (2017). Geography markup language. [Online]. Available: http://www.opengeospatial.org/standards/gml

[18] OGC. (2017). Web map service. [Online]. Available: http://www.opengeospatial.org/standards/wms

[19] OGC. (2017). CityGML. [Online]. Available: http://www.opengeospatial.org/standards/citygml

[20] Player. (2011). Laser format. *doxygen 1.7.4.* [Online]. Available: http://playerstage.sourceforge.net/doc/Player-cvs/player/group__player__driver__writelog__laser.html

[21] Online Management Group. (2017). Domain level task force: Robotics. [Online]. Available: http://robotics.omg.org/

[22] JSA. (2017). Webdesk. [Online]. Available: https://webdesk.jsa.or.jp/

[23] National Institute of Standards and Technology. (2017). National standard certification integrated information system. [Online]. Available: http://standard.go.kr/

[24] ROS. (2017). Navigation package summary. [Online]. Available: http://wiki.ros.org/navigation

[25] International Association of Oil and Gas Producers. (2017). EPSG geodetic parameter registry. [Online]. Available: https://www.epsg-registry.org/

Francesco Amigoni, Politecnico di Milano, Italy. E-mail: francesco.amigoni@polimi.it.

Wonpil Yu, Electronics and Telecommunications Research Institute, Daejeon, South Korea. E-mail: ywp@etri.re.kr.

Torsten Andre, Infineon Technologies Austria AG, Villach. E-mail: Torsten.Andre@ieee.org.

Dirk Holz, University of Bonn, Germany. E-mail: dirk.holz@ieee.org.

Martin Magnusson, Örebro Universitet, Sweden. E-mail: martin.magnusson@oru.se.

Matteo Matteucci, Politecnico di Milano, Italy. E-mail: matteo.matteucci@polimi.it.

Hyungpil Moon, Department of Mechanical Engineering, Sungkyunkwan University, Korea. E-mail: hyungpil@skku.edu.

Masashi Yokotsuka, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan. E-mail: yokotsuka-masashi@aist.go.jp.

Geoffrey Biggs, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan. E-mail: geoffrey.biggs@aist.go.jp.

Raj Madhavan, Humanitarian Robotics Technologies, LLC, Clarksburg, Maryland. E-mail: raj.madhavan@ieee.org.