GAZEBO FINAL TASK

Gazebo we can create a virtual world and load simulated versions of our robots in it. The simulated sensors detect the environment and publish the data.
Gazebo can convert URDF to SDF automatically.

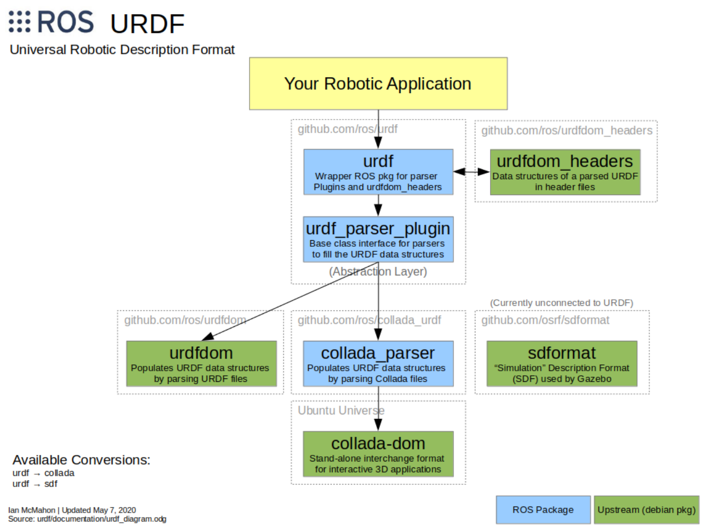
SDF

SDF stands for Simulation Description Format. It is a complete description for everything from the world level down to the robot level. You can accurately describe all aspects of a robot using SDFormat, whether the robot is a simple chassis with wheels or a humanoid. In addition to kinematic and dynamic attributes, sensors, surface properties, textures, joint friction, and many more properties can be defined for a robot. These features allow you to use SDFormat for both simulation, visualization, motion planning, and robot control. Simulation requires rich and complex environments in which models exist and interact. SDFormat provides the means to define a wide variety of environments.

URDF

It is an extensible markup language (XML) file type that includes the physical description of a robot. It is essentially a 3-D model with information around joints, motors, mass, etc. The URDF files are then run through the Robot Operating System (ROS). The data from the URDF file informs the human operator what the robot looks like and is capable of before they begin operating the robot.
It stands for **Unified Robotics Description Format.** It specifies the kinematic and dynamic properties of a robot in isolation.

RDF syntax breaks proper formatting with heavy attributes which in turns makes URDF more inflexible
No mechanism for backward compatibility. Backward compatible (also known as downward compatible or backward compatibility) refers to a hardware or software system that can successfully use interfaces and data from earlier versions of the system or with other systems.
XML format for representing a robot model. Modular robots can be easily included into simulation tools and the effort required for changing one or more modules reduces significantly.


ROS URDF
Universal Robotic Description Format
Your Robotic Application
github.com/ros/urdf
urdf
Wrapper ROS pkg for parser Plugins and urdfdom_headers
github.com/ros/urdfdom_headers
urdfdom_headers
Data structures of a parsed URDF in header files
urdf_parser_plugin
Base class interface for parsers to fill the URDF data structures
(Abstraction Layer)
(Currently unconnected to URDF)
github.com/ros/urdfdom
urdfdom
Populates URDF data structures by parsing URDF files
github.com/ros/collada_urdf
collada_parser
Populates URDF data structures by parsing Collada files
github.com/osrf/sdformat
sdformat
"Simulation" Description Format (SDF) used by Gazebo
Ubuntu Universe
collada-dom
Stand-alone interchange format for interactive 3D applications
Available Conversions:
urdf → collada
urdf → sdf
Ian McMahon | Updated May 7, 2020
Source: urdf/documentation/urdf_diagram.odg
ROS Package
Upstream (debian pkg)

There is now a distinction between a URDF file and a URDF data structure. A URDF file follows the XML format as described on the ros.org wiki. A URDF data structure is a set of generic classes that various formats (currently URDF and Collada) can be parsed into.

Building the file

In order to create a URDF file, we define our robot, ie, define its specifications. URDF is ultimately a tree structure with one root link. This means that the leg's position is dependent on the base_link's position. If we define two structures with the same origin, they will overlap each other. If we don't specify a rpy (roll pitch yaw) attribute, the child frame will by default have the same orientation as the parent frame. We can use the material tag to change the texture/ color of our robot model.

Making the robot move

The connection between the different figures is a continuous joint, meaning that it can take on any angle from negative infinity to positive infinity. We provide additional information like the axis of rotation. There are two other kinds of joints that move around in space. Whereas the prismatic joint can only move along one dimension, a planar joint can move around in a plane, or two dimensions. Furthermore, a floating joint is unconstrained, and can move around in any of the three dimensions.

URDF in Gazebo

To link Gazebo and ROS, we specify the plugin in the URDF. Now that ROS and Gazebo are linked, we specify some bits of ROS code that we want to run within Gazebo, which are called controllers. For every non-fixed joint, we need to specify a transmission, which tells Gazebo what to do with the joint. For every non-fixed joint, we need to specify a transmission. To use a

URDF file in Gazebo, some additional simulation-specific tags must be added to work properly with Gazebo. You can use the same geometry or meshes for both your collision and visual elements, though for performance improvements we strongly suggest you have a simplified model/meshes for your collision geometry. The joint element characterizes the robot from a kinematic point of view, by presenting the attributes *xyz* and *rpy*, which describe the translation and rotation components of joint frame *i* from the joint frame *i* − 1 of the previous element. The input and output connectors are defined for each module. The position and orientation of these connectors are crucial for the correct kinematics definition of the robot, but since they do not add degrees of freedom to the robotic structure, they could be seen as fixed joints. For this reason, it is necessary that during the design, both link and joint modules are modeled as an assembly. When the URDF is obtained, all the properties of the first link element are arranged with respect to the global origin of the software.

DIFFERENCES

URDF describes the robot, SDF on the other hand describes the environment/ world that is simulated as well as the models inside it. Helps us use models in different worlds and also to reuse the world. It is easier to add/modify elements in an SDF file as compared to an URDF file. SDF is designed to represent a superset of everything that can be represented in URDF. SDF was devised by Gazebo to meet simulation needs, but Gazebo can consume URDF when it is augmented by information within `<gazebo>` tags.

XACRO

It is an acronym for XML MACRO. With XACRO, you can construct shorter and more readable XML files by using macros that expand to larger XML expressions. Useful during robot descriptions. It helps in simplifying the URDF files. XACRO allows you to specify properties which act as constants. It helps in building complex expressions using basic operations. Here, the parameters act just like properties, and you can use them in expressions. The main feature of xacro is its support for macros. Define macros with the macro tag, and specify the macro name and the list of parameters. The list of parameters should be whitespace separated. They become macro-local properties. It provides YAML support.