The file display\_models.launch is a ROS launch file, which is an XML file used by the ROS framework to specify and launch multiple nodes (executable programs) with specific configurations. Let's break down the file:

xml

<launch>

This tag signifies the beginning of a ROS launch file.

xml

<arg name="model" />

This defines an argument named model. This argument is not given a default value, so when you run this launch file, you might need to provide a value for this argument. However, it's notable that this argument isn't used elsewhere in the provided snippet, so it might be a leftover or placeholder for future use.

xml

<arg name="robot\_namespace" default="/"/>

This defines an argument named robot\_namespace with a default value of /. The robot namespace is useful when you have multiple robots or instances and need to avoid naming collisions. For instance, each robot can publish its data under its unique namespace.

xml

<param name="robot\_description" command="$(find xacro)/xacro '$(find limo\_description)/urdf/limo\_ackerman.xacro' robot\_namespace:=$(arg robot\_namespace)" />

This line sets a ROS parameter named robot\_description. The value of this parameter is obtained by processing the limo\_ackerman.xacro file found in the limo\_description package. The xacro tool is used to parse xacro files (XML macros) into plain URDF (Unified Robot Description Format) XML. The robot\_namespace argument is passed as a parameter to this xacro file, which might use it to namespace certain parts of the robot description.

xml

<node name="joint\_state\_publisher" pkg="joint\_state\_publisher\_gui" type="joint\_state\_publisher\_gui" ></node>

This line launches the joint\_state\_publisher\_gui node from the joint\_state\_publisher\_gui package. This node provides a GUI to publish joint states, allowing you to manually set and adjust joint values, which is useful for visualization and testing.

xml

<node name="robot\_state\_publisher" pkg="robot\_state\_publisher" type="robot\_state\_publisher" />

This line launches the robot\_state\_publisher node from the robot\_state\_publisher package. This node listens for joint states (published by the previous node or other sources) and uses the robot\_description parameter to compute and publish the 3D poses of each part of the robot, allowing tools like RViz to visualize the robot.

xml

<node name="rviz" pkg="rviz" type="rviz" args="-d $(find limo\_description)/rviz/model\_display.rviz" />

This line launches the rviz visualization tool from the rviz package with a specific configuration file (model\_display.rviz). This configuration file is found in the limo\_description package under the rviz folder. RViz will use the published robot state and the robot\_description to visualize the robot.

xml

</launch>

This tag signifies the end of the ROS launch file.

**How it interacts with other files**:

* **limo\_ackerman.xacro**: This file (found in the limo\_description/urdf directory) describes the robot's structure, joints, links, and visual/collision properties. The launch file processes it to set the robot\_description parameter.
* **model\_display.rviz**: This file (found in the limo\_description/rviz directory) contains the RViz tool's configuration settings for visualizing the robot.

In essence, this launch file is set up to visualize the robot described in the limo\_ackerman.xacro file using RViz. It provides a GUI to manually adjust the robot's joints and visualize how the robot looks in different configurations.