

Group 2

Members: Kelsey Alysha D'Souza kad2185 - Robert Kwiatkowski rjk2147 - Yan-Song Chen yc3240

1. (1) `src` contains catkin source code, consisting of the top-level `CMakeFiles.txt` list of `cmake` instructions, along with all static source files, scripts, and libraries contained in packages. (2) The build folder, or build space, is where `CMake` build is invoked and stores intermediate files and cache information. (3) `devel` stores executables, libraries, `setup.sh`, and other build results prior to installation (generated and compiled outputs). If targets are directly built here, they can be run without installing, e.g. for testing purposes. (4) The log space stores package build logs, named using `{cmd/verb}.{stage}.{index}.log` convention. (5) The install folder, or install space, stores the `CMakeFiles.txt` specified “`install(...)`” targets, with `.pc` package configuration files for package linking instructions.
2. What is a ROS node? A process using ROS data types to communicate with other node processes. The robot control system is a graph comprised of many nodes, the base of which is invoked by “`roscore`” which provides the ROS Master for directing node communication.
3. What is a ROS package? Analogous to a python module, each ROS package aims to provide an API with self-contained functionality. Each package contains its own `CMakeFiles.txt` and any node and message instructions, along with a `package.xml` declaring any package dependencies and containing package details.
4. ROS Master coordinates and directs node requests via XML-RPC protocol. The ROS Parameter Server is a global nested dictionary of `<ROS/path/to/param>: <XML-RPC data type value>`, read and written to by nodes. The nested quality comes from being able to store and retrieve dictionaries of parameters from children namespaces. The `rosout` node subscribes the `rosout` topic for recording and broadcasting log messages.
5. ROS topics are used for continuous data streaming between multiple senders and receivers, with the rate of dataflow determined by the publisher. Services are blocking calls and should be used for short RPCs which terminate quickly, are side-effect free, and are state-independent. Actions are non-blocking processes which direct side effects, discrete robot behaviors which can be preempted and monitored by state.
6. (1) The parameter `hardware_launch/use_gui` is a boolean set to `false` initially. (2) `catkin` is another package used by this system, specified in the topmost `CMakeLists.txt`. (3) The node `model_rec2` is defined in the `vision` launch file and runs on start.

References:

<http://wiki.ros.org/roslaunch>

<http://wiki.ros.org/ROS/>

<http://wiki.ros.org/pcl/Overview>

<http://catkin-tools.readthedocs.io/en/latest/mechanics.html>

https://answers.ros.org/question/253704/what-does-catkin_make-do-during-package-building/