ROS

1 Introdução

O progresso da comunidade robótica na constituição de CPS (*Cyber-physical Systems*) vem aumentando a necessidade por construção de algoritmos que ajudam no comportamento dos robôs nos mais diversos graus de autonomia. O rápido progresso na pesquisa resultou no crescimento da coleção de bons algoritmos para tarefas comuns como navegação, planejamento de movimentos, mapeamento entre outros. A existência destas coleções só se torna útil com a possibilidade de sua reutilização em diferentes contextos sem a necessidade de reimplementação de cada algoritmo em cada sistema.

Vários são os aspectos que devem ser considerados na concepção de algoritmos para sistemas robóticos. Dentre estes pode-se citar (a) a complexidade dos algoritmos, que pode ser dividido em partes que se comunicam para alcançar determinado objetivo, (b) a interoperabilidade entre robôs para realização de uma tarefa compartilhada, (c) concorrência entre processos do sistema que precisa ser analisado e (d) o tratamento de comandos enviados pelo operador em tempo real para os robôs a partir de uma estação em terra. O ROS, sigla para o termo em inglês *Robotic Operating System*, é uma plataforma construída que visa facilitar este processo. É definido como um meta sistema operacional que provê serviços esperados de um sistema operacional, incluindo abstração de *hardware*, controle de baixo nível dos periféricos, implementação de funcionalidades comumente usadas, passagem de mensagem entre processos e gerenciamento de pacotes. Além disso, ele provê ferramentas e bibliotecas para obter, construir, escrever e executar código entre múltiplos computadores [2]. Desta forma, criando uma padronização e agilidade no desenvolvimento de sistemas robóticos a partir da reutilização de componentes especializados já bem estruturados e da universalização da abordagem para integração dos sistemas envolvidos.

2 Estrutura

Todo o ROS é organizado em pacotes. Um pacote ROS é uma coleção coerente de arquivos, geralmente incluindo executáveis e arquivos de suporte, com um determinado objetivo. Tais pacotes podem ser instalados, removidos e listados. Cada pacote tem um arquivo manifesto associado que descreve suas características.

A plataforma possibilita que o roboticista execute mais de um programa, pertencente ao seu respectivo pacote, ao mesmo tempo. Num ambiente de execução, cada programa é chamado de nó. Para orquestração, o ROS utiliza um esquema de troca de mensagens através de publicação e assinatura que registra os serviços dos nós em execução de forma que todos os nós se comuniquem. Tal orquestrador é chamado de *ROS master*. Para que um programa em execução seja considerado efetivamente um nó, *ele deve implementar a lógica de comunicação do ROS master*

O mecanismo de comunicação primário entre nós é o envio de mensagens. As mensagens são organizadas em tópicos de forma que o nó que quer compartilhar informação para outro nó publicará em um tópico. Já o nó que quer receber tais informações irá assinar aquele tópico. O *ROS master* cuida para que publicadores e assinantes achem um ao outro, porém, as mensagens são enviadas diretamente do publicador para o assinante [1].

Referências

[1] J. M. O'Kane. *A Gentle Introduction to ROS*. CreateSpace Independent Publishing Platform, 10 2013.

[2] D. Thomas. ROS Introduction. http://wiki.ros.org/ROS/Introduction, 2014. Acessado 06/03/2015.