Robot Operating System (ROS)

Walter Fetter Lages

fetter@ece.ufrgs.br

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG10052 Laboratório de Robótica

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Introdução

ROS é um pseudo sistema operacional com ferramentas para desenvolvimento de *software* para robôs:

- Gerenciamento de pacotes
- Abstração de *hardware*
- Bibliotecas com algoritmos comumente utilizados
- Simuladores
- Mecanismos de comunicação
- *scripts* úteis

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Porque ROS?

- Código aberto
- Centralização das informações
- Reuso de código
- Desenvolvimento em grupo
- Processamento inerentemente distribuído
- Nodos fracamente acoplados
- Parar de reinventar a roda

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Histórico

- Sistema desenvolvido em Stanford em 2000 para o robô STAIR 1
- Aperfeiçoado em 2007 pela Willow Garage para o robô PR2 e denominado ROS

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Versões do ROS 1

- Noetic Ninjemys - 23 de maio de 2020
- Melodic Morenia - 23 de maio de 2018
- Lunar Loggerhead - 23 de maio de 2017
- Kinetic Kame - 23 de maio de 2016
- Jade Turtle - 23 de maio de 2015
- Indigo Igloo - 22 de julho de 2014
- Hydro Medusa - 4 de setembro de 2013
- Groovy Galapagos - 31 de dezembro de 2012
- Fuerte Turtle - 23 de abril de 2012
- Electric Emys - 30 de agosto de 2011
- Diamond Back - 2 de março de 2011
- C Turtle - 2 de agosto de 2010
- Box Turtle - 2 de março de 2010

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Sistema Operacional *Host*

- Linux é o sistema operacional *host*
- A distribuição "oficial" é a Ubuntu
- Outras distribuição suportadas:
  - Ubuntu ARM
  - OS X
  - Yocto
  - Debian
  - Arch Linux
  - Ângström
  - UDOO
  - Android
  - Código fonte
  - robotpkg

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Conceitos

- O ROS é organizado em pacotes
  - Nodos (processo do sistema operacional hospedeiro)
  - programas utilitários
  - Bibliotecas
  - Definições de mensagens
  - Arquivos de configuração
  - *Plugins*
- Pacotes podem ser agrupados em metapacotes

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Pacotes

- Menor nível na organização do ROS
- Dedicados a uma única funcionalidade
- Cada pacote deve ser implementado em um diretório

Estrutura de um Pacote ROS 1

```
catkin_ws/
├── build/
│   └── nome_do_pacote/ ..... Temporários
├── devel/
│   ├── lib/ ..... Bibliotecas
│   ├── nome_do_pacote/ ..... Executáveis
│   └── setup.bash ..... Script de configuração
└── src/
    └── nome_do_pacote/
        ├── src/ ..... Códigos-fonte
        ├── launch/ ..... Scripts de carga
        ├── scripts/ ..... Outros scripts
        ├── include/ ..... Arquivos de cabeçalho
        ├── package.xml ... Metadados e dependências
        └── CMakelist.txt .. Configuração do Cmake
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Metapacotes

- Agrupam pacotes que em conjunto oferecem uma funcionalidade mais abstrata
- Metapacotes são pacotes "vazios"
  - Servem para agrupar outros pacotes como dependências
  - Baixando um metapacote, se baixa todas as dependências
  - Pacotes não devem depender de metapacotes, apenas de pacotes

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

roscore

- `roscore` é um conjunto de nodos e programas que são pré-requisitos para um sistema ROS 1
- Deve ser executado para que os nodos do ROS 1 possam se comunicar
- É lançado com o comando `roscore`
- O comando `roslaunch` também lança o `roscore`, se ele ainda não estiver executando
- O `roscore` inicia os seguintes nodos:
  - ROS master
  - ROS parameter server
  - `rosout`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ROS Master

- Nodo que provê serviços de registro e consulta de nomes de nodos, tópicos e serviços
- Ao iniciar, os nodos devem registrar-se com o mestre
- Ao subscrever um tópico, os nodos consultam o mestre e estabelecem conexão diretamente entre si
- É lançado pelo comando `roscore`
- O computador onde o *ROS Master* está executando é especificado através da variável de ambiente `ROS_MASTER_URI`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Servidor de Parâmetros

- Nodo que provê um dicionário acessível aos demais nodos
- É lançado pelo comando `roscore`
- Nodos podem usar o servidor de parâmetros para armazenar e recuperar parâmetros
- Não projetado para alto desempenho
  - Adequado para parâmetros de configuração
  - Não é um substituto para aplicações de banco de dados

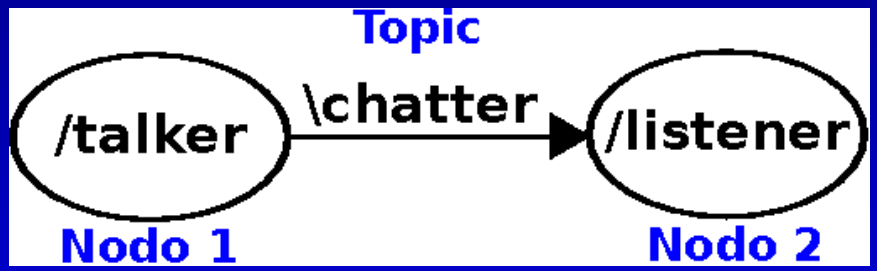
Comunicação através de Tópicos

Nodo: processo do S.O. hospedeiro

Tópico: mecanismo de comunicação entre nodos do tipo *publisher/subscriber*

Mensagem: dados publicados nos tópicos

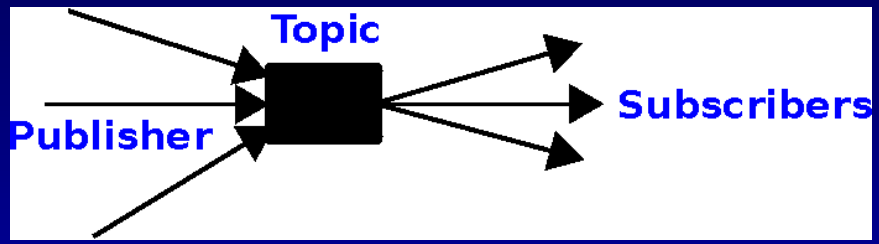
Gráfico de computação: Representa a comunicação entre os nodos através de tópicos



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Tópicos

- Nodos podem publicar mensagens em tópicos
- Cada tópico pode ter vários publicadores e assinantes

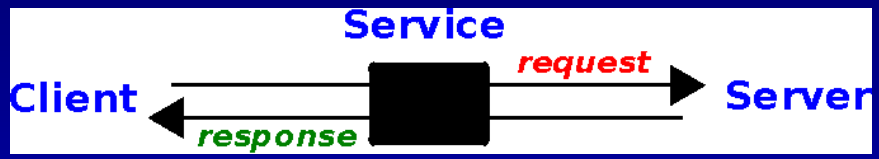


- Cada nodo pode publicar ou assinar vários tópicos
- Publicadores e assinantes não sabem da existência um dos outros
- A ordem de execução não é garantida
- Comunicação assíncrona

Serviços

- Tópicos não são apropriados para solicitação de serviços entre nodos

Serviço: mecanismo de comunicação entre nodos do tipo *remote procedure call* (RPC)

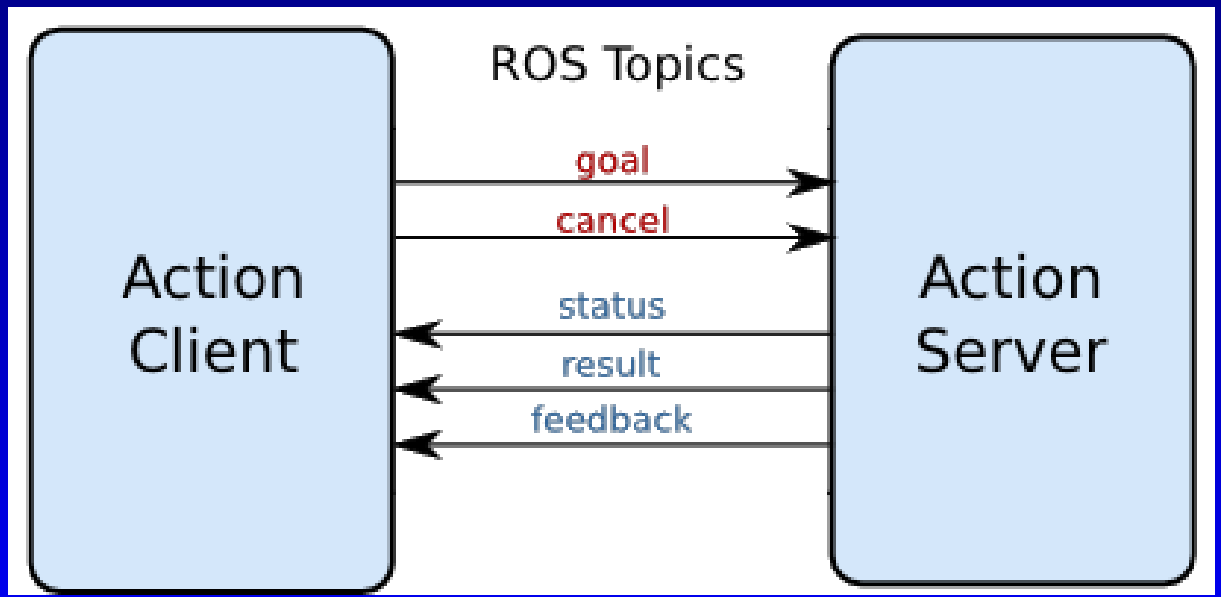


- Serviços oferecem um mecanismo de requisição/resposta

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Action Servers

- Apropriados para serviços exigem longo tempo de execução
- Permitem o cancelamento da requisição
- Permitem receber informações sobre o *status* da execução



Rviz

- Rviz é uma ferramenta do ROS para vizualização
- Mostra de forma conveniente dados publicados em tópicos

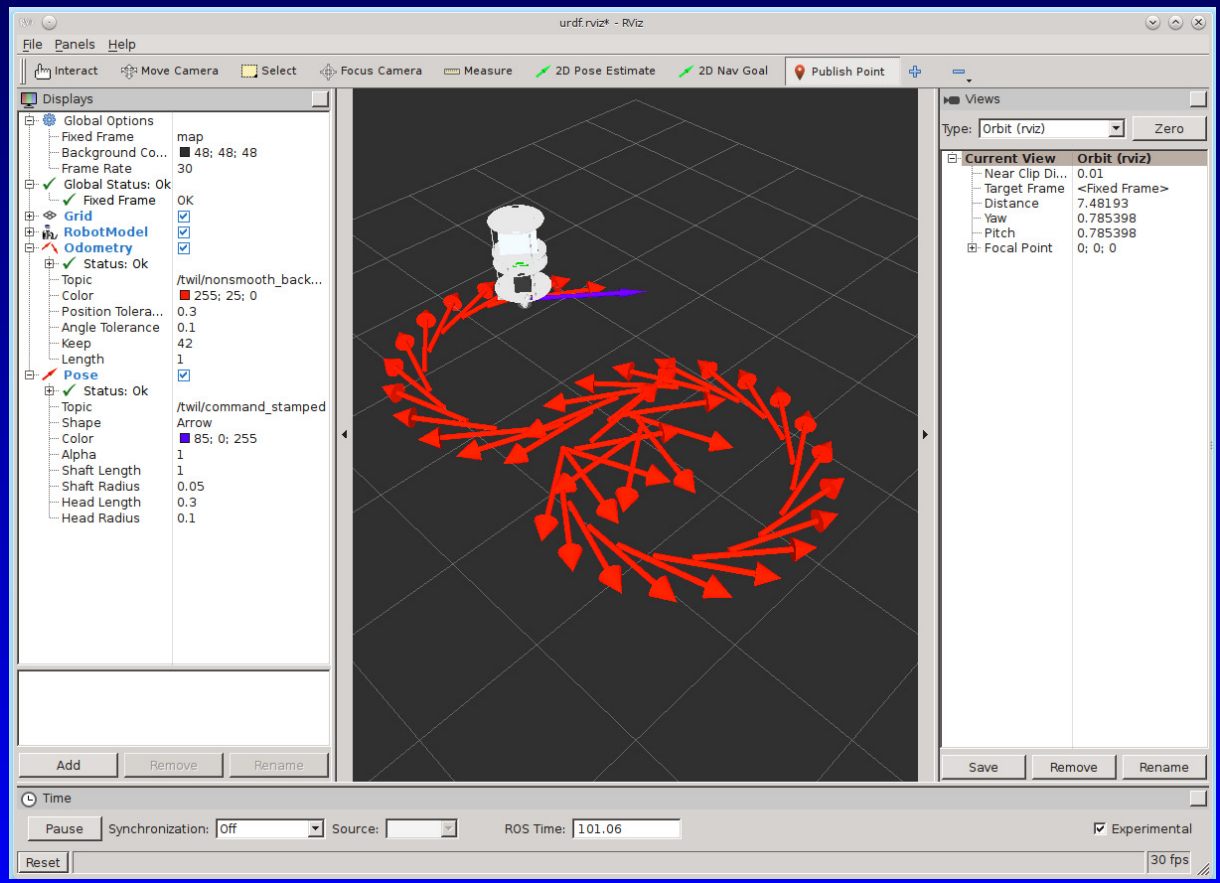
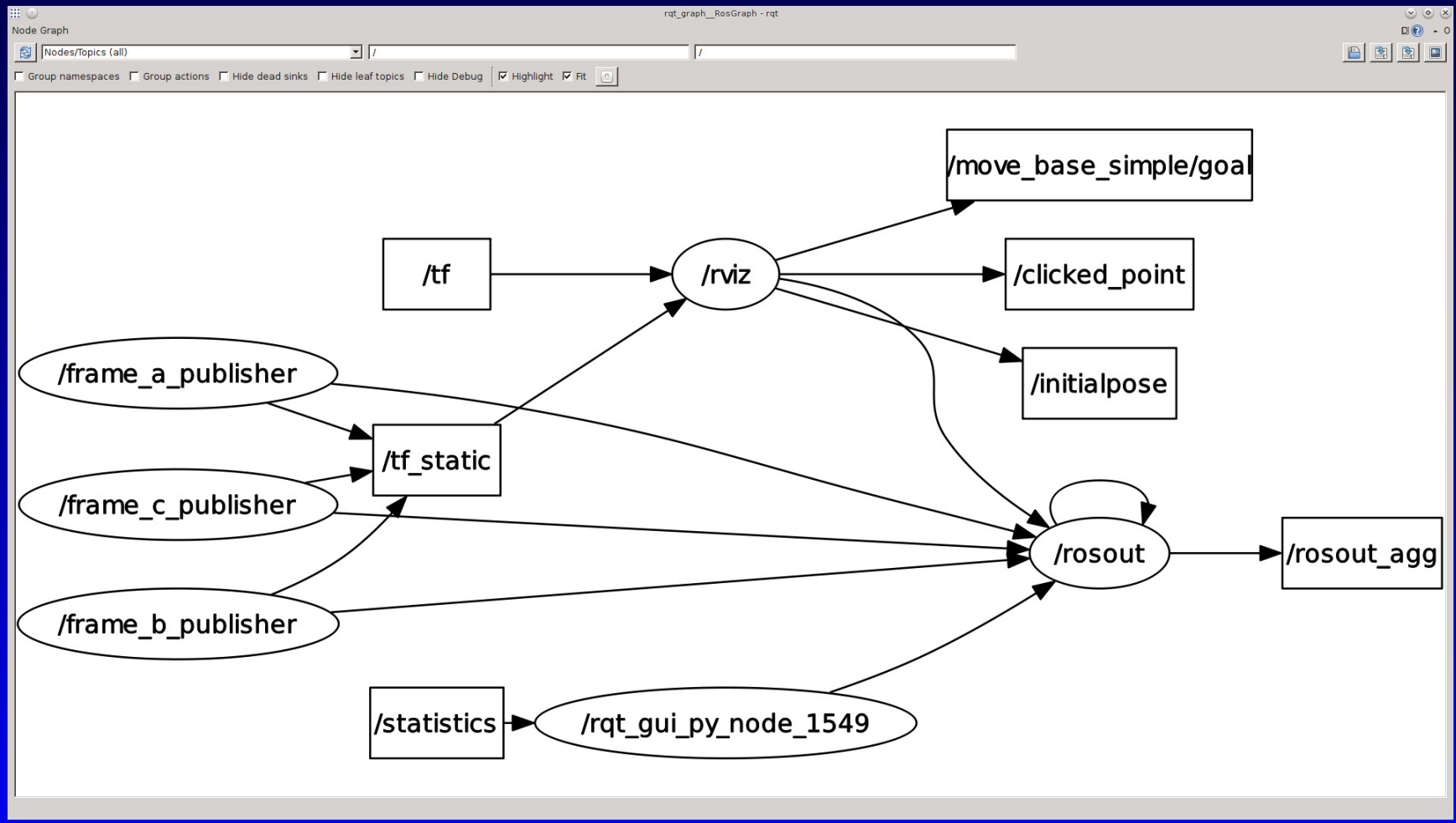


Gráfico de Computação

- A ferramenta `rqt_graph` Permite vizualizar os nodos e tópicos

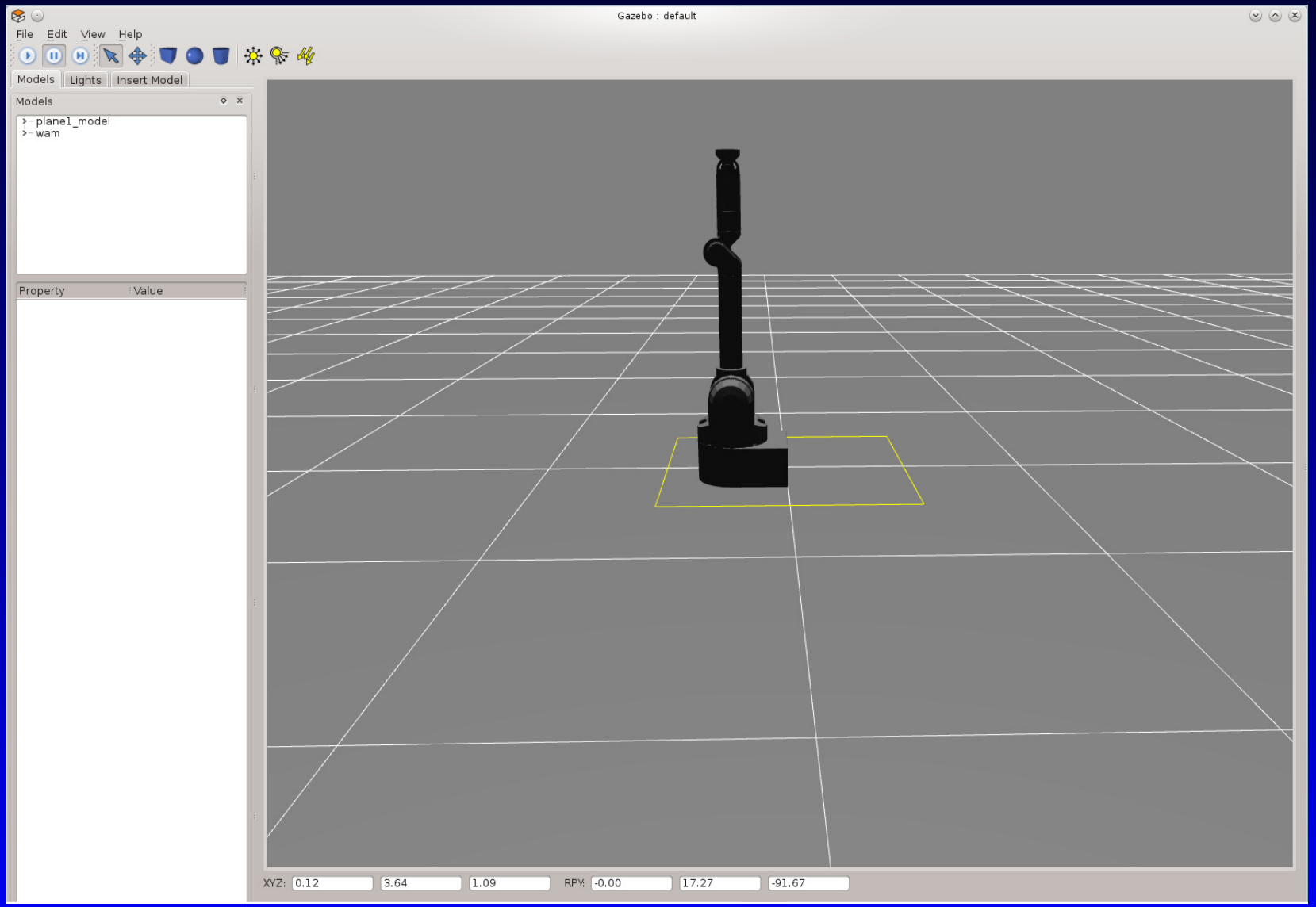


UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

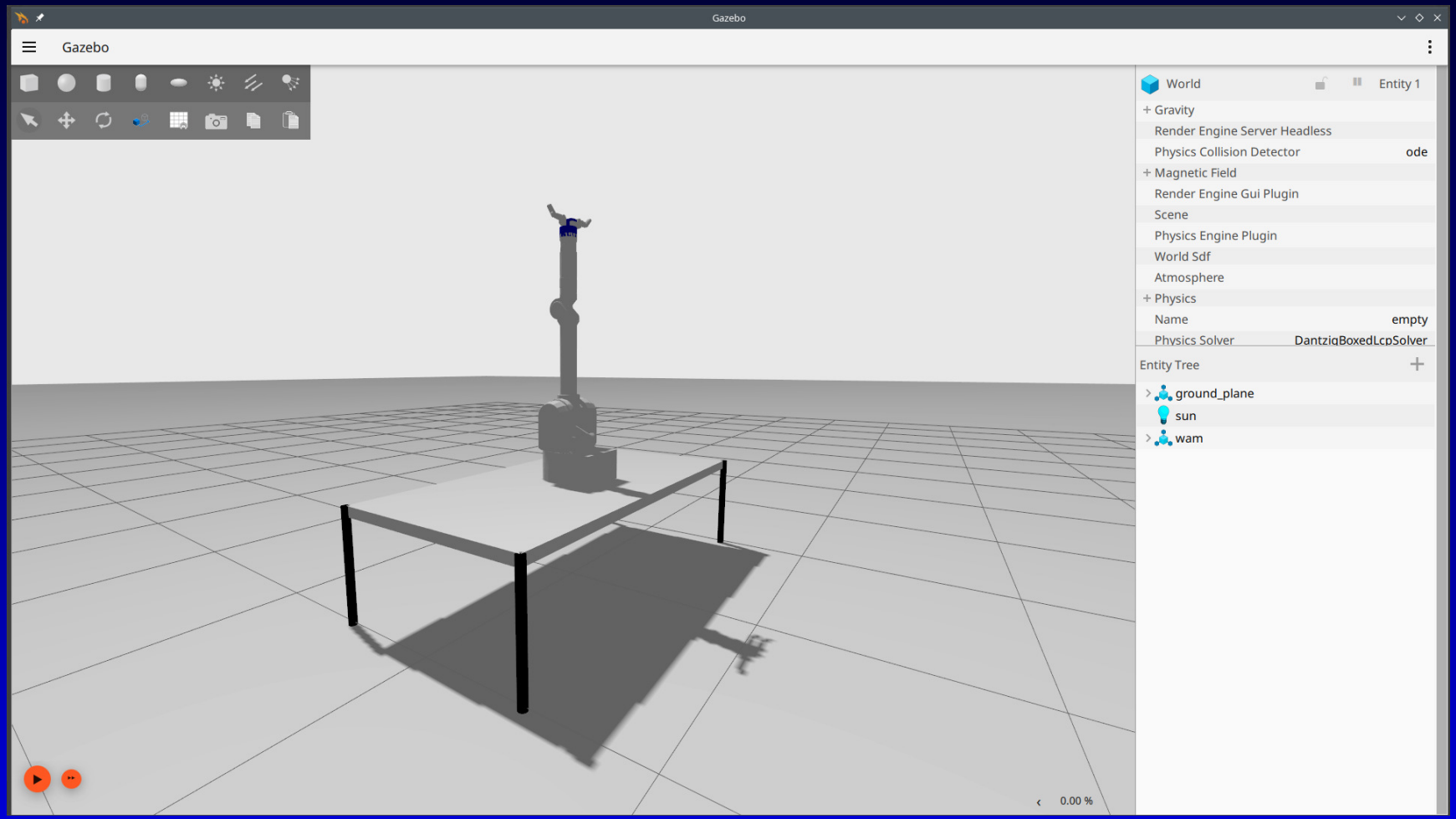
Gazebo

- Simulador 3D
- Suporta diversos *backends*
  - *Open Dynamics Engine* (ODE)
  - *Bullet*
  - *Dynamic Animation and Robotics Toolkit* (DART)
- O modelo do robô é descrito em URDF
- Gazebo: até a versão 11, descontinuado
  - Gazebo Classic
- Ignition: versão 6 em diante
  - Renomeado para Gazebo
  - Gazebo Ignition

Barrett WAM no Gazebo



Barrett WAM no Gazebo Ignition



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ROS 2

- O ROS foi criado como um sistema de desenvolvimento para o PR2
  - Suporta apenas um robô
  - Recursos computacionais abundantes *on board*
  - Sem requisitos de tempo-real
  - Excelente conectividade
  - Aplicações em pesquisa
  - Máxima flexibilidade (sem *warp* do `main()`)
- O ROS 2 procura solucionar esses problemas

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Porque ROS 2?

- Novos casos de uso: times de robôs, plataformas embarcadas, sistemas de tempo-real, rede não ideal, ambientes de produção (ROS-I)
- Novas tecnologias: Zeroconf, Protocol buffers, ZeroMQ, Redis, WebSockets, Data Distribution Service (DDS)
  - Já é possível construir o *middleware* com componentes de prateleira
- Mudanças na API
  - Muitas mudanças poderiam instabilizar o ROS1

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Diferenças do ROS 2

- Testado nas plataformas Ubuntu, e Windows 10
- Cada pacote tem a sua configuração de ambiente
- Apenas *isolated build* é suportado
- Sem diretório devel, pacotes precisam ser instalados para serem usados
- *Namespaces* separados para `.msg` e `.srv`
- Campos primitivos em mensagens podem ter valores *default*
- Tipos `duration` e `time` unificados
- Cabeçalhos de mensagens sem o campo `seq`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Diferenças do ROS 2

- Não é preciso executar o `roscore`
- Nodos tem um ciclo de vida que pode ser gerenciado
- Suporte para tempo-real (se o sistema operacional suportar)
- Arquivos de *launch* escritos em Python, XML ou YAML
- Pacotes binários para o Windows baseados no Chocolatey

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Versões do ROS 2

- Rolling Ridley - desde junho 2020
  - Rolling release
- Jazzy Jalisco - 23 de maio de 2024
- Iron Irwini - 23 de maio de 2023
- Humble Hawksbill - 23 de maio de 2022
- Galactic Geochelone - 23 de maio de 2021
- Foxy Fitzroy - 5 de junho de 2020
- Eloquent Elusor - 22 de novembro de 2019
- Dashing Diademata - 31 de maio de 2019
- Crystal Clemmys - 14 de dezembro de 2018
- Bouncy Bolson - 2 de julho de 2018
- Ardent Apalone - 8 de dezembro de 2017

Estrutura de um Pacote ROS 2

```
colcon_ws/
├── build/
│   └── nome_do_pacote/ ..................Temporários
├── install/
│   └── nome_do_pacote/
│       ├── lib/ ..................Bibliotecas e executáveis
│       ├── share/ ..................Configuração e launch
│       ├── local_setup.bash ..Script de configuração local
│       └── setup.bash .........Script de configuração global
├── log/ ..........................................logs
└── src/
    └── nome_do_pacote/
        ├── src/ ..........................Códigos-fonte
        ├── launch/ .....................Scripts de carga
        ├── scripts/ ....................Outros scripts
        ├── include/ ..............Arquivos de cabeçalho
        ├── package.xml .......Metadados e dependências
        └── CMakelist.txt .......Configuração do Cmake
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Instalação

- Sistema operacional *host*
  - Linux
    - Ubuntu 22.04 (Kubuntu 22.04)
      - Pacotes Debian
      - Arquivo *fat*
    - Red-Hat Enterprise Linux
      - Pacotes RPM
      - Arquivo *fat*
  - Windows 10
  - macOS
  - Instação a partir do código-fonte
    - Deve funcionar em qualquer Linux

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Instalação

- Instalação do Linux
  - Dual-boot
  - Máquina virtual
    - Problemas com algumas placas gráficas
  - Windows System for Linux (WSL)
    - Problemas com algumas placas gráficas
  - Docker
- Instalação do ROS 2: `http://docs.ros.org`
  - Versão Humble
  - Instalada a partir de pacotes Debian
    - Instalada em /opt/ros/humble