


ROS入門講座（第一部）

本講座では、主に「ROSとは何か？」に注目し、技術的な側面ではなく、ROSを活用するために必要な背景知識の獲得を目標としている。ROS環境を用いたプログラム開発等に興味がある方は、本講座の最後に示すチュートリアル等を実施してもらおうか、本NEDO特別講座で開講している他の講座を聴講して欲しい。

NEDO特別講座編集

2021年5月公開バージョン

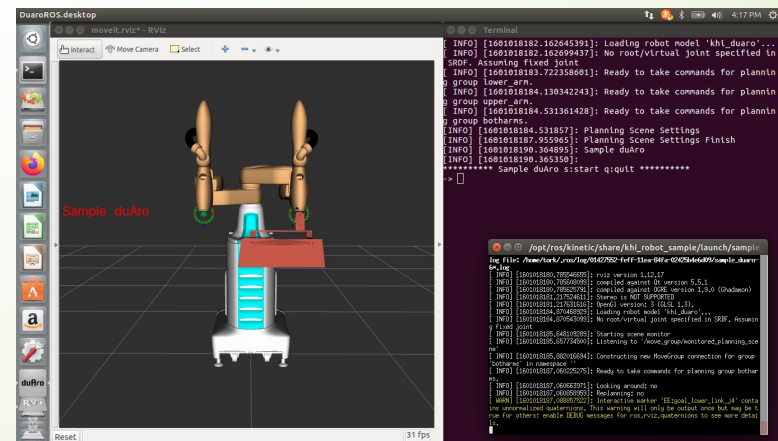
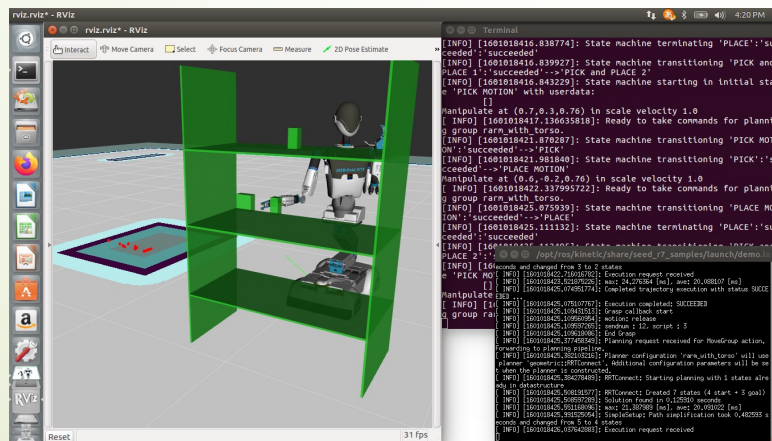


この講座で学ぶこと

- ➡ ROSとは何か？
- ➡ なぜ、ROSなのか？
- ➡ ミドルウェア技術、OSS活用とは？
- ➡ ROSの活用事例、協力企業
- ➡ ROSの抱える問題点
- ➡ ROSの基本技術概要（コンポーネントの紹介）
- ➡ プログラム開発を自分で学ぶためのリソース

ROSとは何か？

- ROS「Robot Operating System」とは、ロボットを活用するためのソフトウェア環境です
- 大きな流れとして、当初からの開発の流れでできてきたROS1と、そこから派生したROS2が存在します
 - 現在は新しいコンポーネント間通信を備えたROS2の開発が進んでいます。が、有用なライブラリがすべて移植されているわけではないので、ここではROS1を対象とした講座とします



OSなのですか？

- ➡ **はい、と言っても、あまり問題ないかと思えます**
 - ➡ ROS1は、Ubuntu Linuxに付加することによって動作するソフトウェア環境となっており、基本的にはLinuxと呼ばれるOSと不可分です。OSと同様のサービスをユーザーの作るアプリケーションに提供します
- ➡ **現代的な用語を用いれば、ミドルウェア、あるいはソフトウェアフレームワーク、プラットフォームと呼ぶ方が正確です**
 - ➡ ROS1には、様々な機器のデバイスドライバやロボットのナビゲーション、マニピュレーションを実現するライブラリだけでなく、視覚化ツール、メッセージ通信、パッケージ管理のツールも含まれています

Windows上で動きますか？


- ➡ ROS1では、仮想環境 (Docker) を用いて動作している例もありますが、一般のアプリケーションのように動作させることはできません
 - ➡ ROS1は、原則、Ubuntu Linuxが必須と考えておいたほうがよいでしょう
- ➡ ROS2では、普及促進の観点から複数のOS (Windows, mac os) 上での動作を目指して開発が進められています。

補足：現在、WSLと呼ばれる、LinuxアプリケーションをWindows上で稼働させる仕組みの開発が急速に進められており、プロトタイプ段階ですがROSと後述するGazeboシミュレータを用いたWindows上での開発が可能になっています

なぜ、ROSが注目されているのでしょうか？

- **様々な分野でのロボット利活用が加速度的に増加しています。その結果、ロボット関連の技術開発の加速が必要とされています**
- **ソフトウェア開発の効率化の必要性の向上**
 - 共通ソフトウェアの共有による開発工数の低減化
 - 枯れたコードの利用への期待
- **ロボット用ソフトウェア特有の複雑さへの対処**
 - モータ制御、信号処理、軌道計画、様々なハードウェア、など多種多様なプログラムが必要
 - 現実世界との複雑なインタラクションの記述が必要
 - デバックの難しさ：現実世界は時間で変化するため同じ状態（入力）を再現できない
- **最新の高度アルゴリズムの活用の期待**
 - 最新のAI技術、ICT技術の活用に期待

これらを実現できるのがROSなのです



ROSを理解するために必要な キーワード

- ミドルウェア技術, コンポーネント化技術
- フレームワーク, プラットフォーム
- エコシステムとコミュニティ志向の開発
- オープンソースソフトウェア (OSS) 活用

ミドルウェア技術, コンポーネント化技術

▶ ミドルウェア

- ▶ OSとアプリケーションソフトウェアの間に位置し, 様々な共通的な処理を引き受けるソフトウェア群

▶ コンポーネント

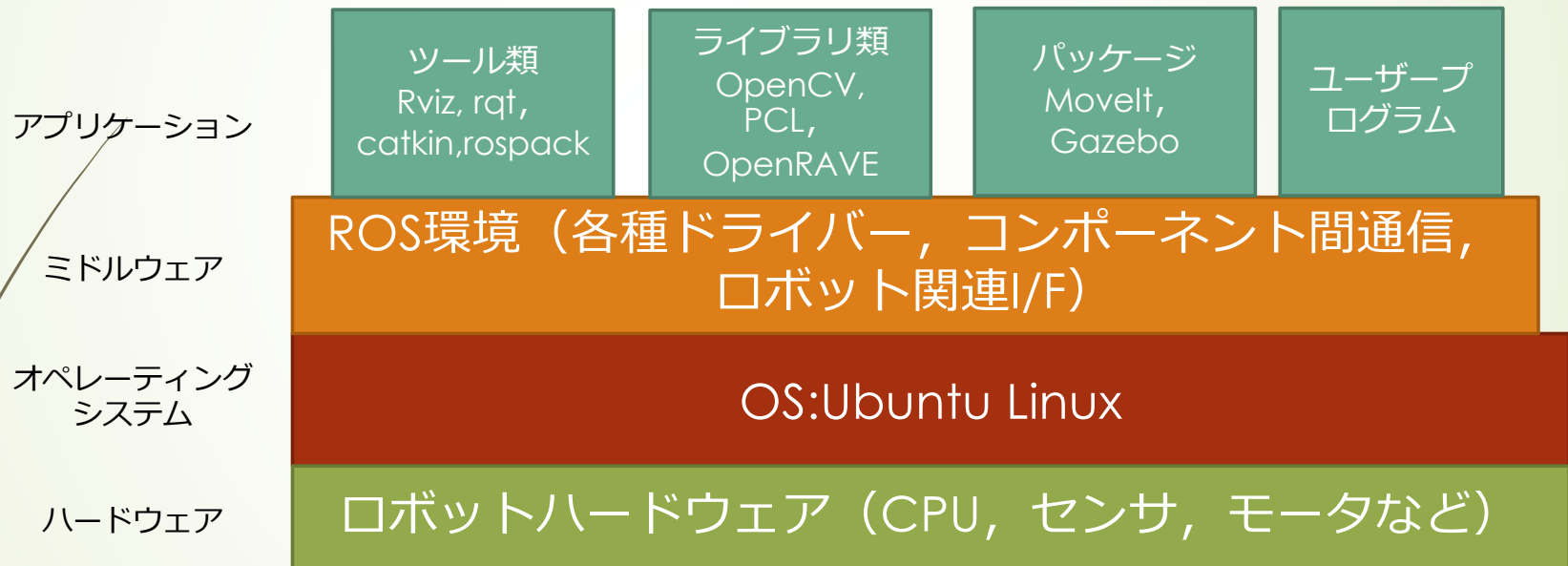
- ▶ ある機能を果たす部品化されたソフトウェア. 従前のライブラリと異なり, 独立性が高い (環境に依存しない等) ことが特徴

▶ なぜ, この二つに注目するのか?

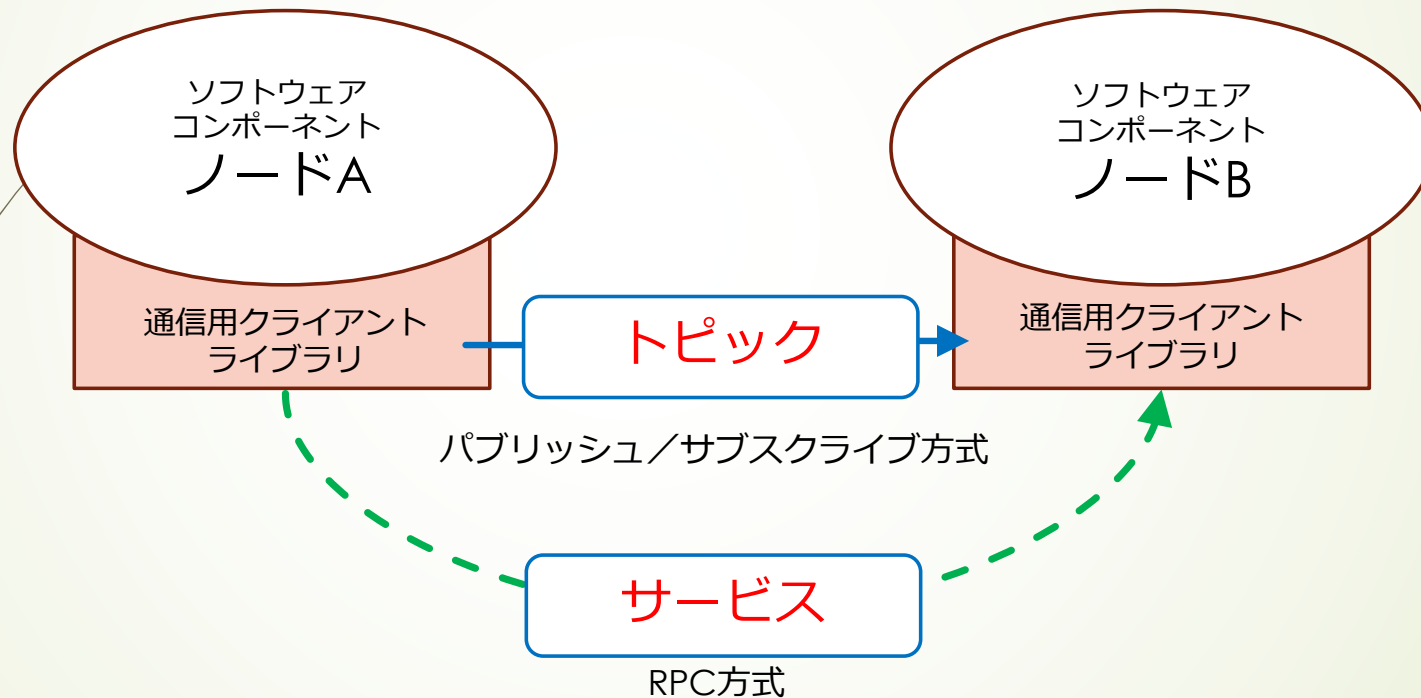
- ▶ 共通するソフトウェアは, 競争領域ではなく, 協調領域
- ▶ 車輪の再発明はしない
 - ▶ 通信分野, データベース分野でミドルウェア技術は, 重要な手段となっています
- ▶ OSのAPIやソフトウェアライブラリではなく, 独立性の高いコンポーネント開発指向が有利 (移植性が高い)
- ▶ 再利用のレベルが高い (プラグ&プレイが可能)

ミドルウェア技術の役割

-ROSのいるところ-

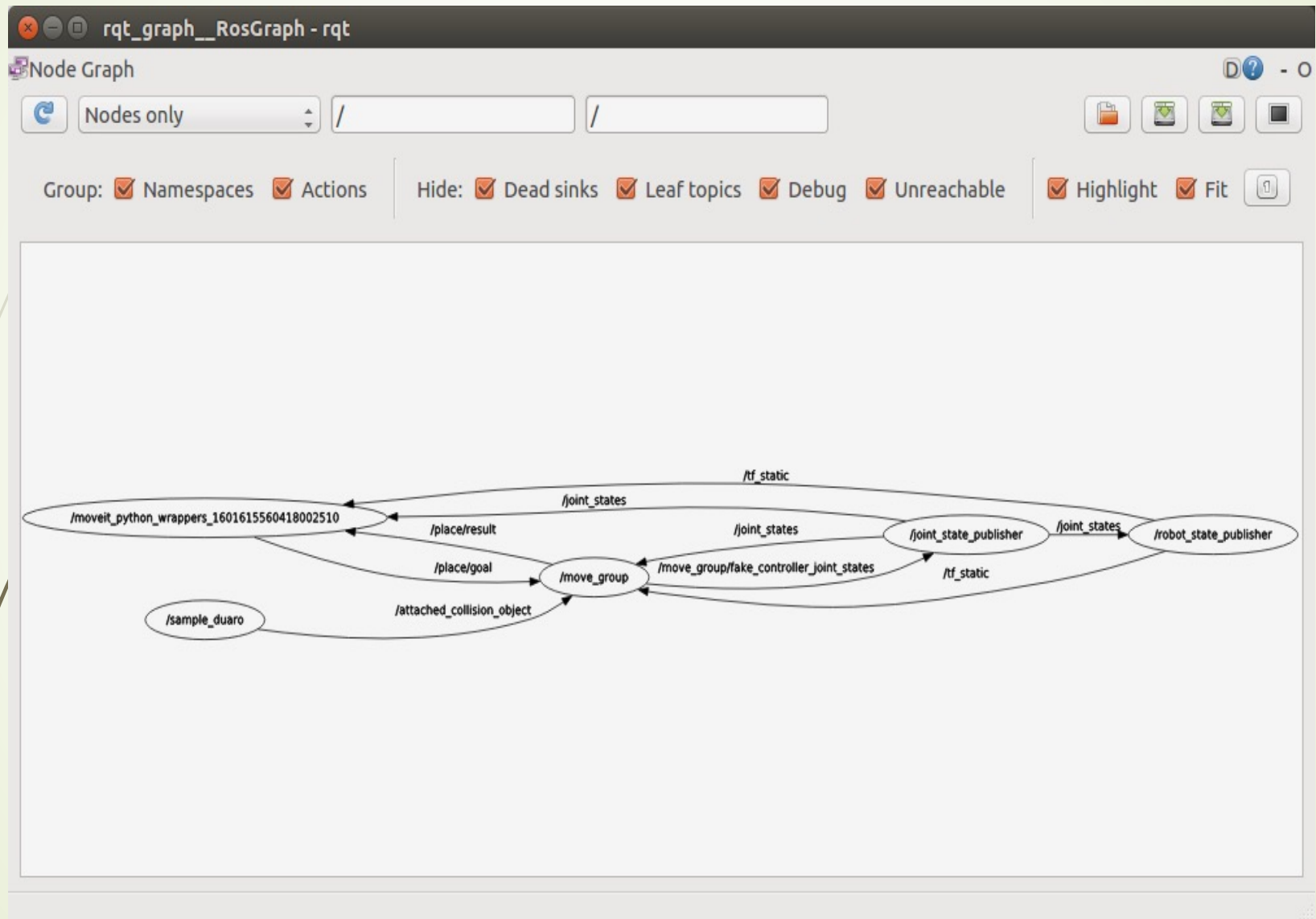


ROSのアプリケーションの基本的な動作



コンポーネント間のデータの送受信を基本として動作
各コンポーネントは、データ処理やハードウェアの制御などを実行

ノードグラフの一例



ROSのツールである「rqt_graph」で出力

OSS活用, コミュニティ指向の開発

▶ OSS(Open Source Software)

- ▶ オープンソースとして開発されたソフトウェア
- ▶ 通常, 無償でソースコードが公開され, 誰でも活用 (利用, 追加, 公開) が可能

▶ OSS活用の例

- ▶ Linux (各種クラウドサーバ, メールサーバ他の大規模, 大容量の基幹システムのOS)
- ▶ 最近では,
 - ▶ Dronecode (ドローン用OS), Autoware (車の自動運転)

▶ コミュニティ

- ▶ 特定のOSSに関わる関係者の集まり. 共同で開発を行ったり, 講習会を実施

ROS開発に投資する企業

- ▶ IT企業やチップメーカーなどが積極的に開発やサポートに取り組んでいます
 - ▶ 例えば、2020年のROS開発者会議のスポンサー企業リストを見ると、その顔ぶれを知ることができます。
<https://roscon.ros.org/world/2020/>
- ▶ IT企業
 - ▶ AWS (Amazon)
 - ▶ Microsoft
 - ▶ Sony. . .
- ▶ チップ企業
 - ▶ Arm
 - ▶ Intel. . .

ROSのコミュニティの規模

- **2020年7月のwiki.ros.orgへのアクセス数（国別）**
 - 中国：3万9千アクセス（20%）
 - 米国：3万1千アクセス（16%）
 - 日本：1万6千アクセス（8%）
 - ドイツ：1万4千アクセス（7%）
- **2020年7月のメインダウンロードサイトからのバイナリパッケージダウンロード数**
 - 8万6千パッケージ（2019年7月時点では、5万5千パッケージ）
 - この数字には、世界各地にあるミラーサイトは含まれていません
- 2020年7月だけで、ディスカッションのサイトには、約2万人が訪れています。2019年に比べて、ディスカッション、質問サイトの訪問者数は、2割から3割増加

* <http://download.ros.org/downloads/metrics/metrics-report-2020-07.pdf> より引用

ROS対応製品

➡ 多種のセンサ類の活用した多様な動作の記述が必要なサービスロボットを中心にROSの活用が進んでいます



Fetch Robotics



ユニバーサルロボットは、世界の50,000を超える製造現場で日々稼働しています

Universal Robots



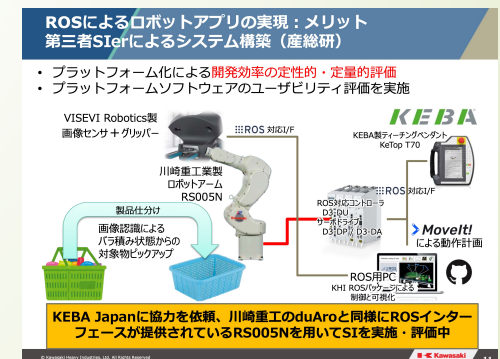
ZMP ロボットカー



4足歩行ロボット SPOT



THK Seed Noid



川崎重工業 ロボットアーム

公式ページ : <https://robots.ros.org>
<http://wiki.ros.org/Sensors>

ROSの抱える問題点

➤ OSSが抱える共通の問題点

- 素早い開発の立ち上がりと最新技術の取り込みが実現できる一方で,
 - ドキュメントの不足, メンテナーの偏在
 - バグが判明したときに, 誰が責任を持って対処してくれるのか?
- 信頼性を担保するための仕組みの不在
- 網羅性の不足
 - 開発者の興味で開発分野が決定されるため, 重要なソフトウェアが未開発だったりする

➤ ライセンス (知財) の問題

- 本NEDO講座の別資料で解説しています

ROS 1 の抱える技術的な問題点

- 1台のみのロボット制御が前提
- 標準化されていない独自の通信環境の実装
- 非リアルタイム
- 単一故障点の存在
- **これらの解決のためのアプローチ**
 - ROS2プロジェクト
 - ROS-I (ROS Industrial)

* ROS2については後で

現在利用可能なROSのディストリビューション

- ▶ ROS1, ROS2の配布用パッケージ（名前が付いています）
- ▶ 最新バージョン+動作環境, サポート期限



ROS1 Melodic

Ubuntu 18.04 LTS
~ 2023年5月



ROS1 Noetic

Ubuntu 20.04 LTS
~ 2025年5月



ROS2 Foxy

Ubuntu 20.04 LTS
~ 2023年5月

* 2021年4月現在

代表的なROSコンポーネント類

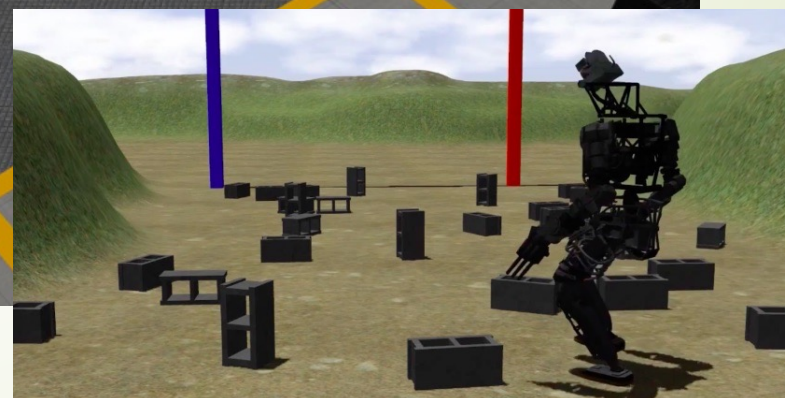
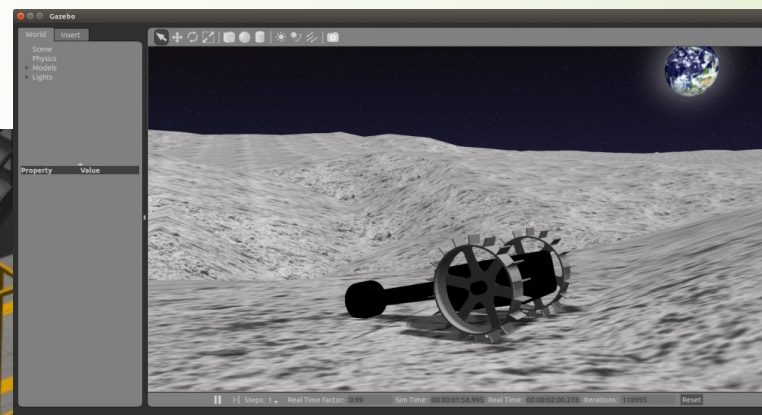
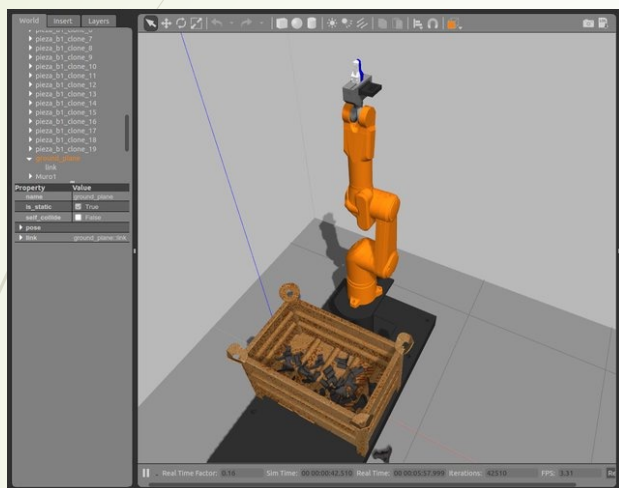
▶ ROSパッケージ

- ▶ 公式サイトに掲載されたものだけでも、2000種類以上
- ▶ パッケージ：パッケージの管理ファイル、ビルドに使用する設定ファイル、ソースコード、ノード間通信のための設定ファイルなどから構成される
 - ▶ センサ関連：深度センサ、カメラ、レーザー距離計等
 - ▶ モーター駆動：様々なロボットのための車輪駆動、関節駆動等
 - ▶ 上位レベルのパッケージ：音声認識、画像処理、画像識別

▶ ツール類

- ▶ Gazebo：ダイナミクスシミュレーター
- ▶ rqt：GUIユーティリティツールの「rqt」
- ▶ RViz：3次元可視化ツール
- ▶ Move It!：マニピュレータの動作制御のための統合ライブラリ
- ▶ ビルドシステムやパッケージのリリースのためのツール

ツール : Gazebo



* ダイナミクスシミュレーター

ツール : RViz



* 3D可視化ツール

コンポーネント : Move It!

moveit.rviz* - RViz

Interact Move Camera Select

Displays

- Global Options
- Global Status: Ok
- Grid
- MotionPlanning
 - Status: Ok

Add Duplicate Remove Rename

MotionPlanning

Context Planning Manipulation Scene Objects Stored Scenes Stored States Status

Planning Library

OMPL

Planner Parameters

<unspecified>

Warehouse

Host: 127.0.0.1 Port: 33829 Connect

Workspace

Center (XYZ): 0.00 0.00 0.00

Size (XYZ): 2.00 2.00 2.00

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Move Z. Shift: More options.

FSRobo ROS Demo

moveit.rviz* - RViz

Interact Move Camera Select

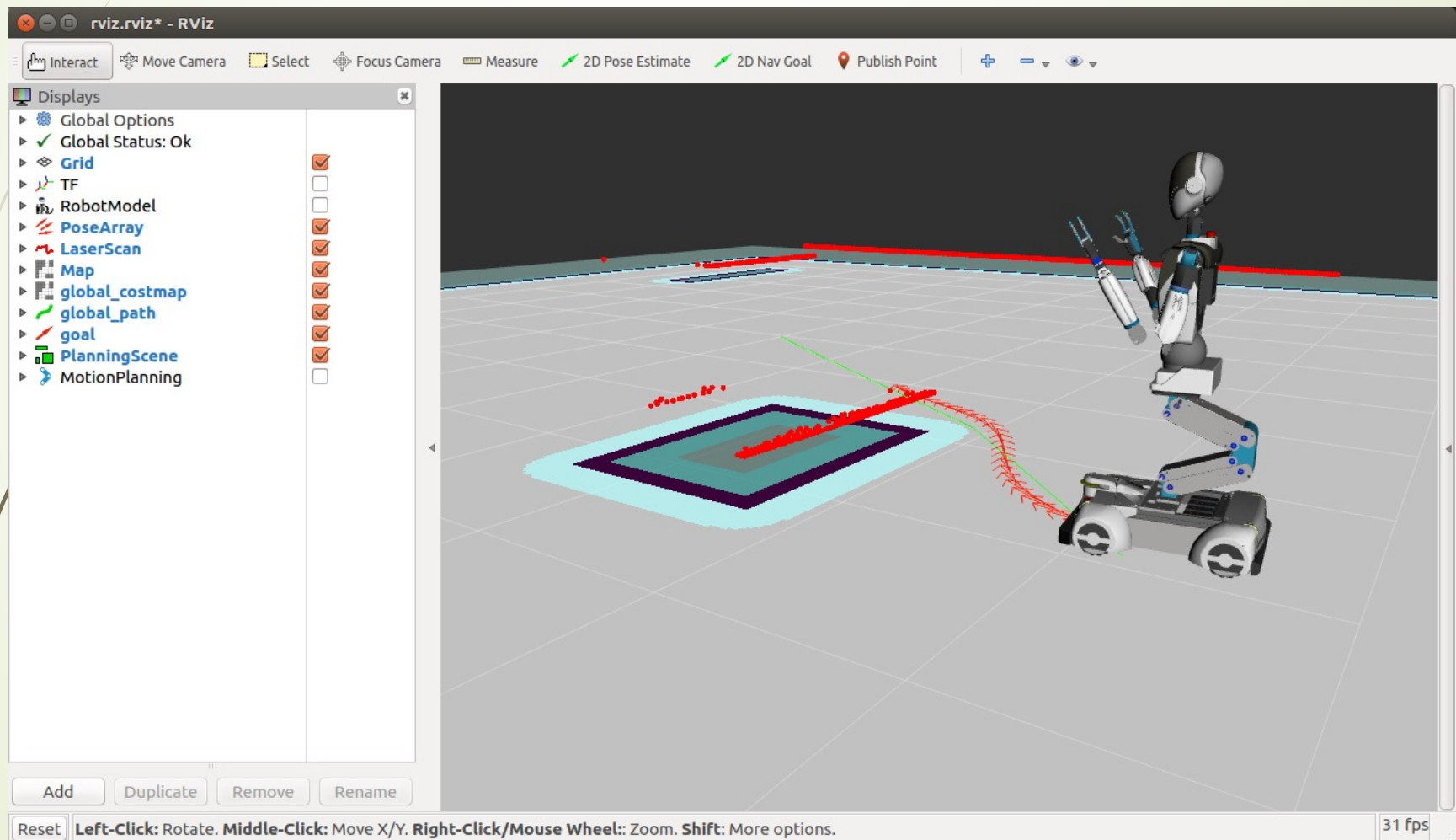
31 fps

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Move Z. Shift: More options.

```
[INFO] [180101820,577029272]: The root link base_link has an inertia specified in the URDF, but KDL does not support a root link with an inertia. As a workaround, you can add an extra dummy link to your URDF.
[INFO] [180101820,577029272]: The root link base_link has an inertia specified in the URDF, but KDL does not support a root link with an inertia. As a workaround, you can add an extra dummy link to your URDF.
[INFO] [180101820,577029272]: Starting scene monitor.
[INFO] [180101820,577029272]: Starting scene monitor.
[INFO] [180101820,577029272]: Listening to '/move_group/monitored_planning_scene'.
[INFO] [180101820,577029272]: Listening to '/move_group/monitored_planning_scene'.
[WARN] [180101820,604072514]: The root link base_link has an inertia specified in the URDF, but KDL does not support a root link with an inertia. As a workaround, you can add an extra dummy link to your URDF.
[INFO] [180101820,607994226]: Constructing new MoveGroup connection for group 'manipulator' in namespace
[INFO] [180101820,607994226]: Ready to take commands for planning group 'manipulator'.
[INFO] [180101820,607994226]: Looking around: no
[INFO] [180101820,607994226]: Replanning: no
[WARN] [180101820,607994226]: Interactive marker 'EtiGoal_Link6' contains unro realized quaternions, this warning will only be output once but may be true for others; enable DEBUG messages for ros_rviz.quaternions to see more details.
```

* ロボットアームの軌道計画

コンポーネント : Navigation Stack



* 移動ロボットのための地図作成, 経路計画



ROS入門講座（第一部）終了