

Robot operációs rendszerek és fejlesztői ökoszisztémák

Movelt

Gincsiné Szádeczky-Kardoss Emese

2025. november 3.

KUKA



iit

Tartalom

Visszatekintés



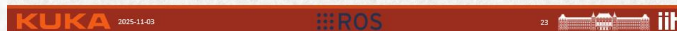
 **Movelt**

Movelt - bevezető



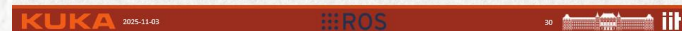
 **Movelt**

Movelt eszközök



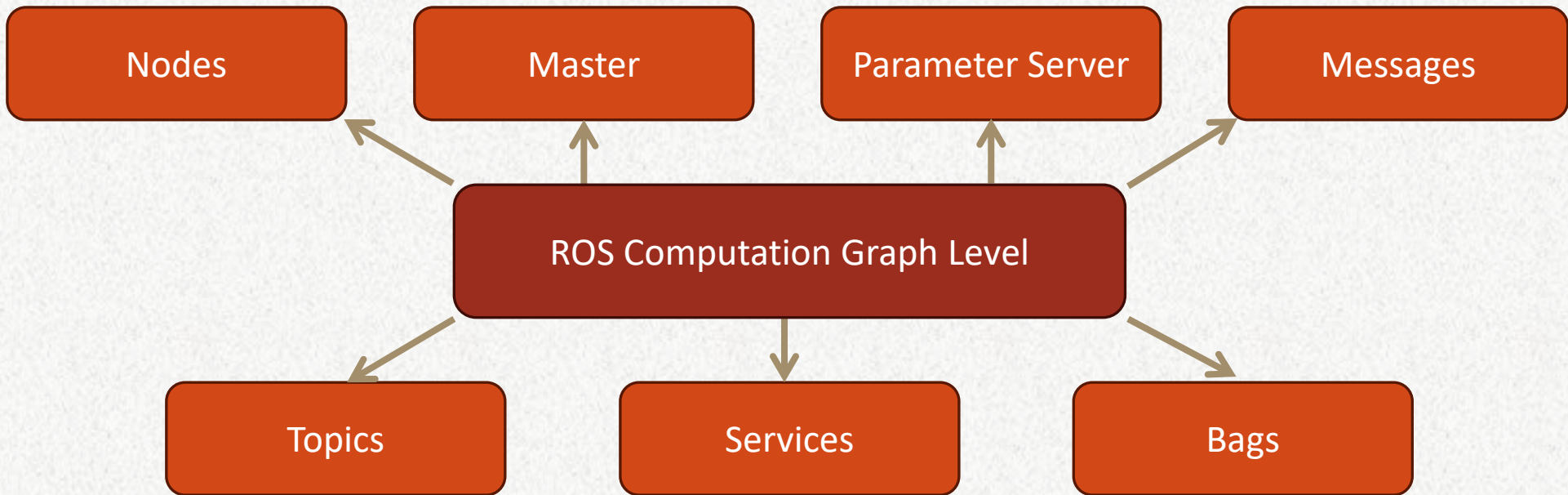
 **Movelt**

Movelt - programozás

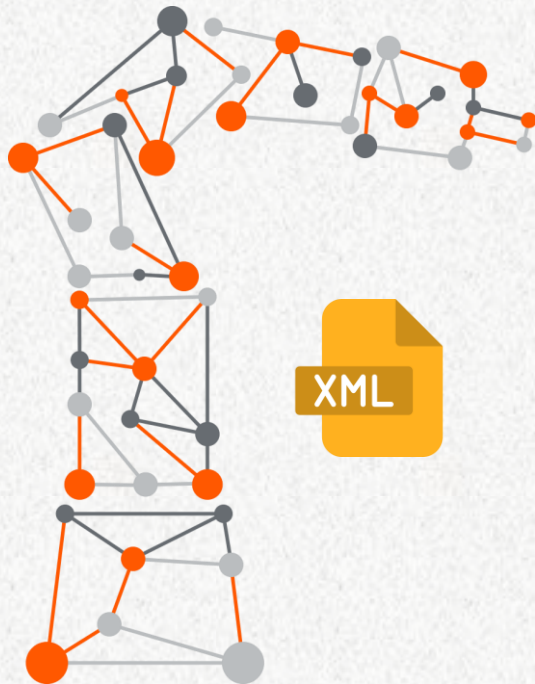


Visszatekintés

ROS bevezető

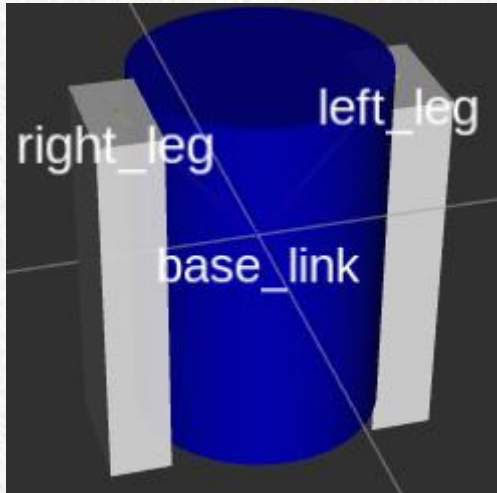


Unified Robotics Description Format



- XML-alapú
- Összekapcsolt merev testek (multibody) modellezése
- Nem része a ROS-nak (más robotikai alkalmazások is használják)
- URDF elemei:
 - `<link>`
 - `<joint>`

URDF - Példa



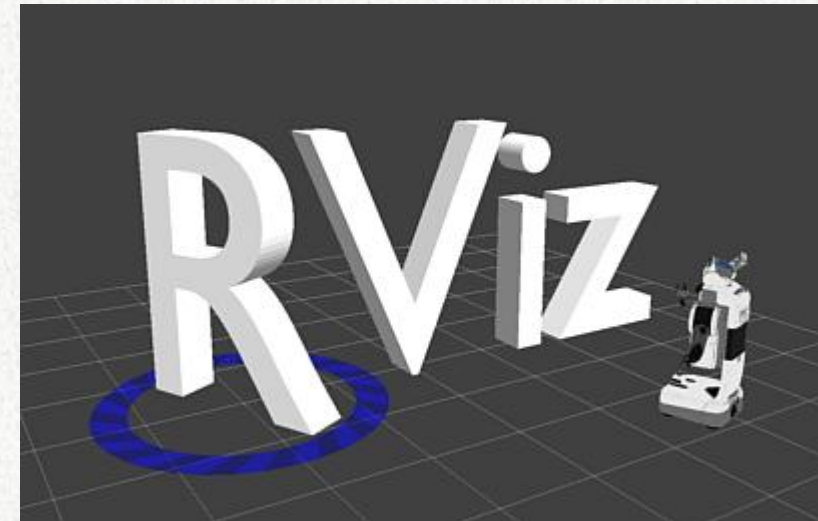
```
1 <?xml version="1.0"?>
2 <robot name="materials">
3
4   <material name="blue">
5     <color rgba="0 0 0.8 1"/>
6   </material>
7
8   <material name="white">
9     <color rgba="1 1 1 1"/>
10  </material>
11
12  <link name="base_link">
13    <visual>
14      <geometry>
15        <cylinder length="0.6" radius="0.2"/>
16      </geometry>
17      <material name="blue"/>
18    </visual>
19  </link>
20
21
```

```
22 <link name="right_leg">
23   <visual>
24     <geometry>
25       <box size="0.6 0.1 0.2"/>
26     </geometry>
27     <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
28     <material name="white"/>
29   </visual>
30 </link>
31
32 <joint name="base_to_right_leg" type="fixed">
33   <parent link="base_link"/>
34   <child link="right_leg"/>
35   <origin xyz="0 -0.22 0.25"/>
36 </joint>
37
38 <link name="left_leg">
39   <visual>
40     <geometry>
41       <box size="0.6 0.1 0.2"/>
42     </geometry>
43     <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
44     <material name="white"/>
45   </visual>
46 </link>
47
48 <joint name="base_to_left_leg" type="fixed">
49   <parent link="base_link"/>
50   <child link="left_leg"/>
51   <origin xyz="0 0.22 0.25"/>
52 </joint>
53
54 </robot>
```

urdf_tutorial/urdf/04-materials.urdf

Vizualizáció

- rqt_gui – ROS integrált GUI
- RViz
 - 3D vizualizációs környezet
 - Megjeleníthető, hogy mit lát a robot, mit gondol a robot, mit csinál a robot:
 - Robot modell
 - Keretek
 - Szenzor információ
 - Markerek
 - ...



Szimuláció

- Beágyazott robot alkalmazás fejlesztése a robot fizikai jelenléte, használata nélkül
- Tervezéséhez, teszteléséhez, tanításához érdemes használni
- Sok előny
- Gazebo:
 - 3D robotszimulátor
 - Fizikai szimuláció (gzserver), grafikus felhasználói felület (gzclient)
 - Szensorok szimulációja
 - Gazebo worlds (*.world), Gazebo models (*.sdf)



ROS-control

- Control: robot egyes jellemzőit változtatva befolyásoljuk, beállítjuk a robot mozgását
 - csukló pozíciók, sebességek, nyomatékok
 - mobilis robot sebessége (twist)
- Szabályozások: beavatkozás a mért kimenet visszacsatolása alapján
- Controller manager, Controller, HW Resource interface, hardware_interface



Movelt - bevezető

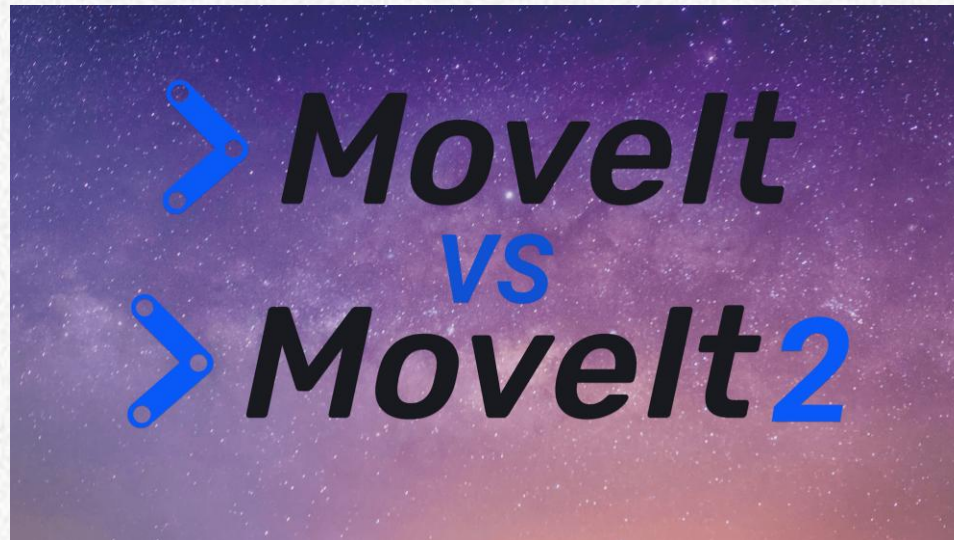
MoveIt

- <https://moveit.ai/>
- MoveIt Motion Planning Framework
 - Robotok mozgástervezése
 - Megfogás tervezés
 - Inverz kinematika: végberendezés kívánt pozíciójából (sebességéből) csuklópólusok meghatározása
 - Robotok irányítás
 - Környezet érzékelés (perception)
 - Ütközés detektálás
- Free, open source



Verziók

- MoveIt (1): https://moveit.github.io/moveit_tutorials/
- MoveIt 2: <https://moveit.picknik.ai/main/index.html>



<https://moveit.ros.org/ros2/moveit/2021/06/08/moveit-vs-moveit2.html>

Megjegyzés: A link egy 2021-es cikkre mutat, azóta már még többet tud a MoveIt2.

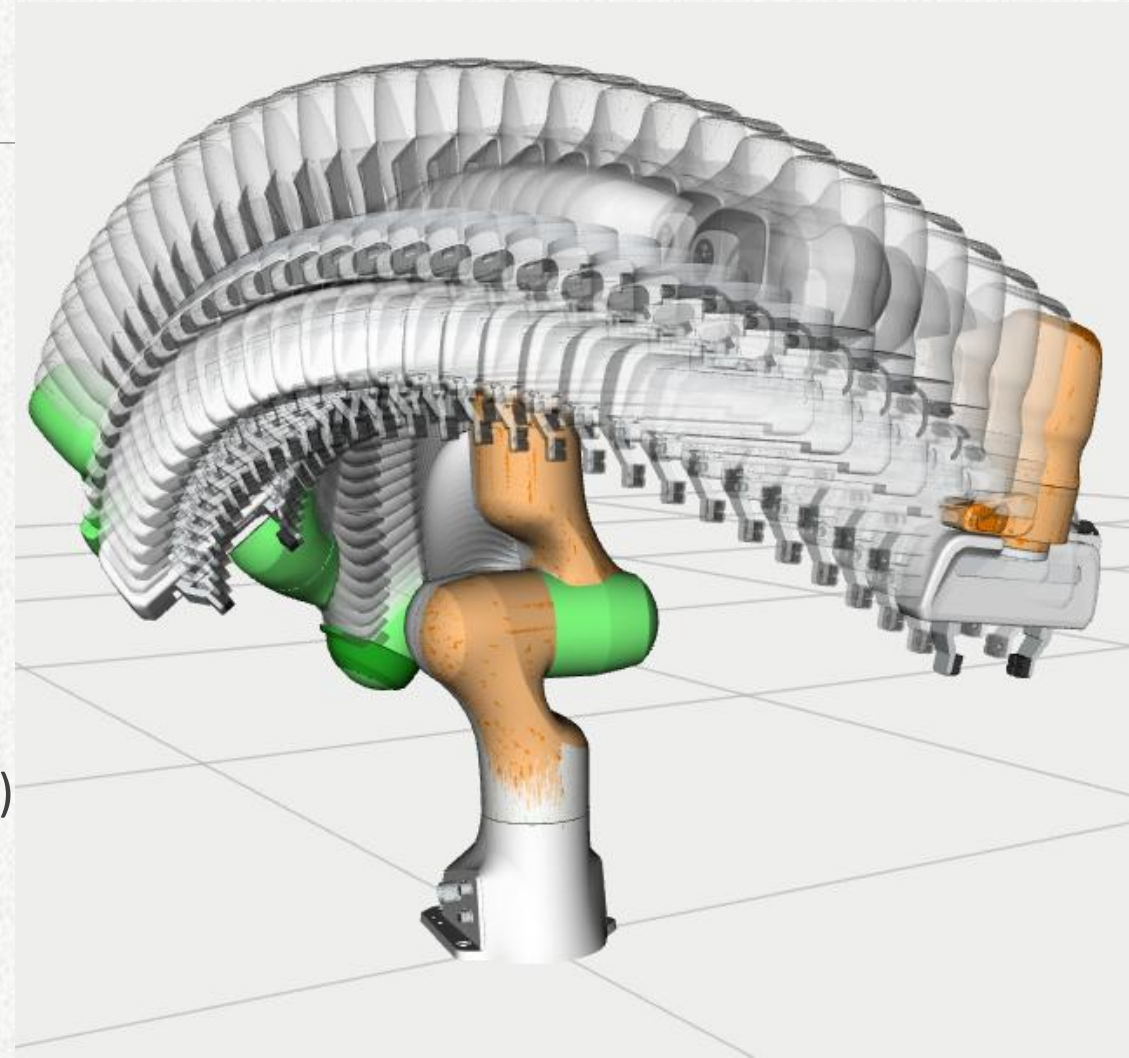
Mozgástervezés

Alap feladat:

- Adott egy kiinduló helyzet
- Adott egy cél
- Keressük olyan megvalósítható robotkonfigurációk sorozatát, amivel a kiinduló helyzetből a célba juttathatjuk a robotkart

Extrák:

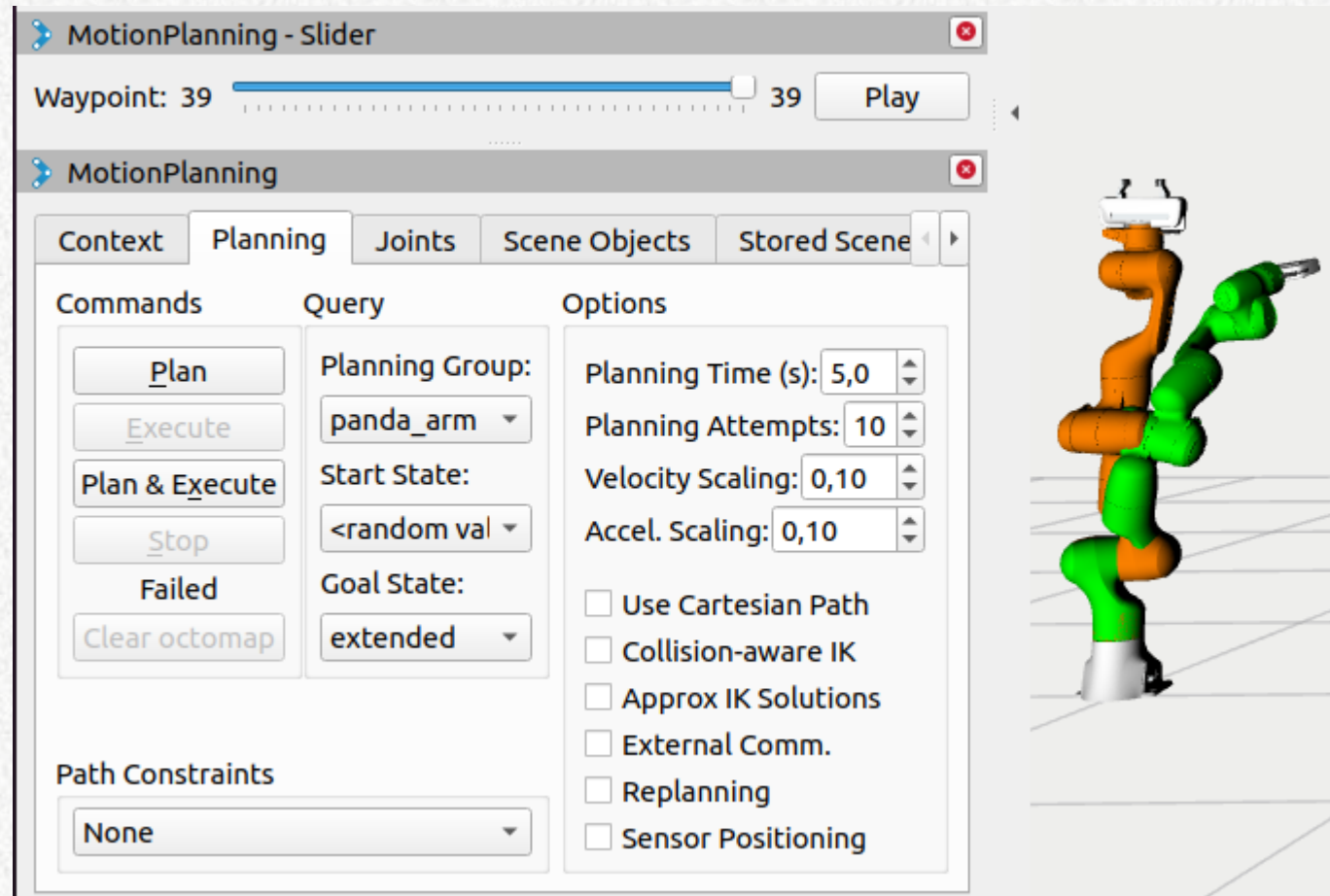
- Ütközésmentes mozgás (környezettel vagy önütközés)
- Optimális mozgás
- Kényszerek, korlátozások (pl. max. gyorsulás)



Ízelítő: MoveIt RViz plugin

Interaktív mozgástervező plugin

- Virtuális környezet beállítása
- Robot kiinduló helyzettének és céljának megadása
 - Többféle opció (interaktív eszköz, random, korábban definiált)
- Többféle mozgástervező alkalmazása
- Mozgástervezés eredményének vizualizálása
- Példa



Konceptió

Rendszer architektúra

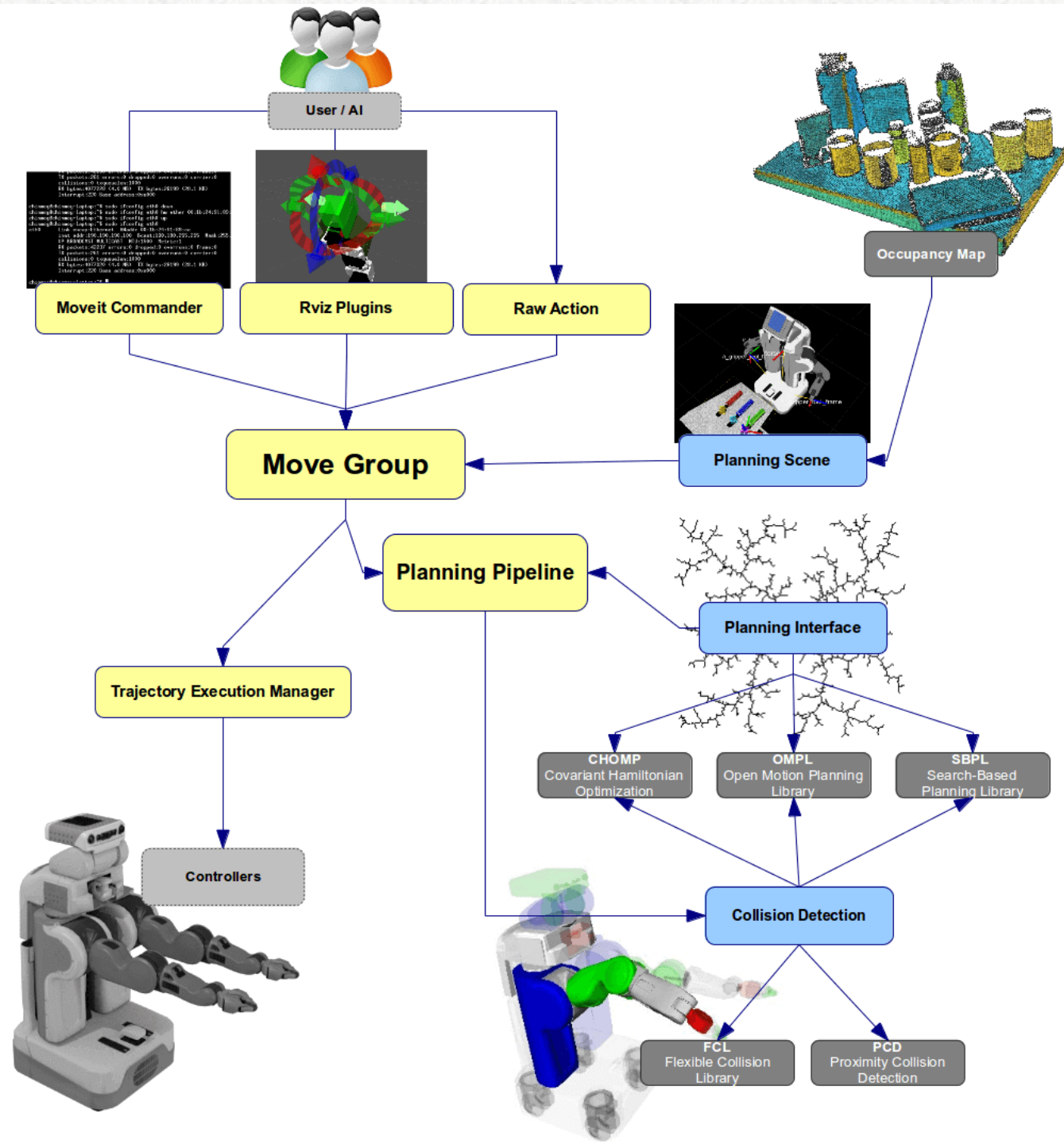
- Fő ROS node: **move_group**
- Plugin-ek (ROS yaml konfigurációs paraméterekkel)

Legend

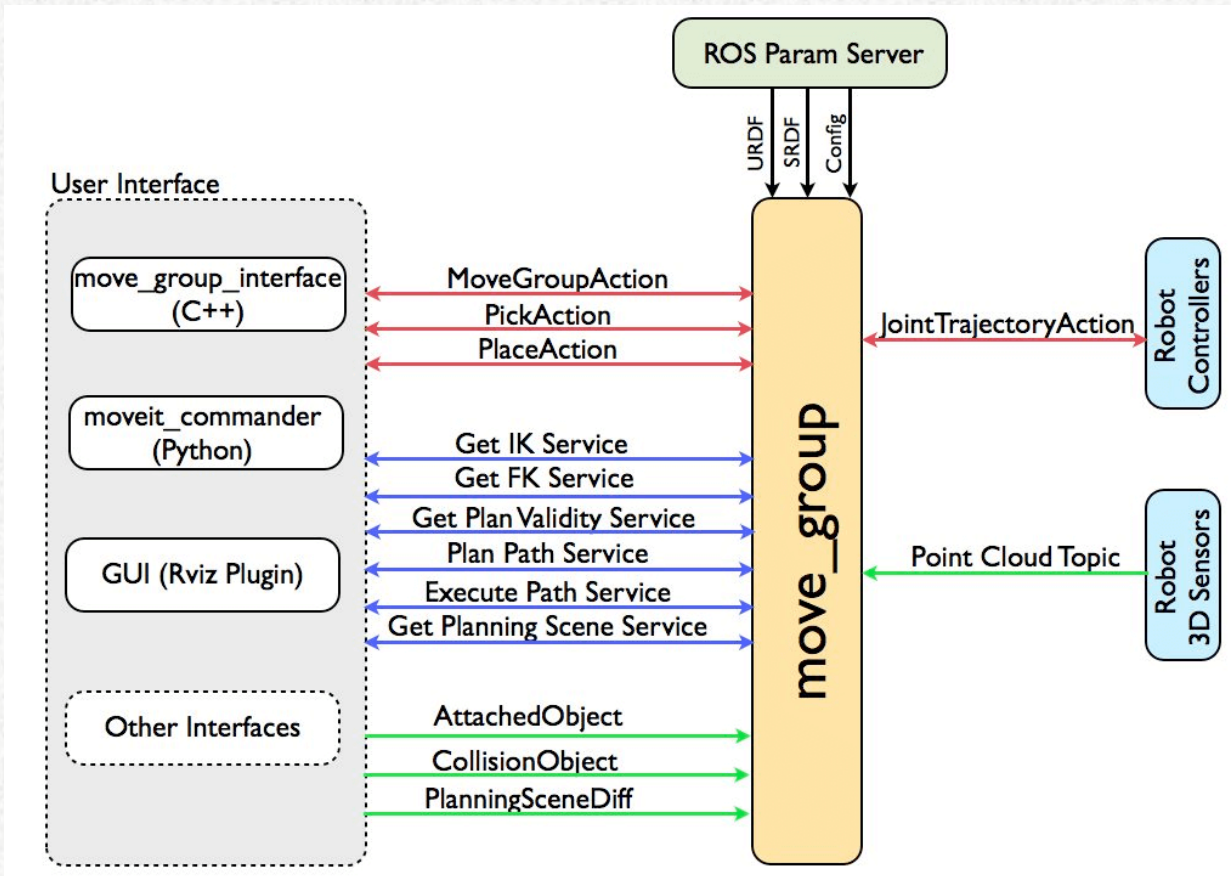
moveit_core
Package

External Package
Dependency

moveit_ros
Package



Konceptió – move_group node



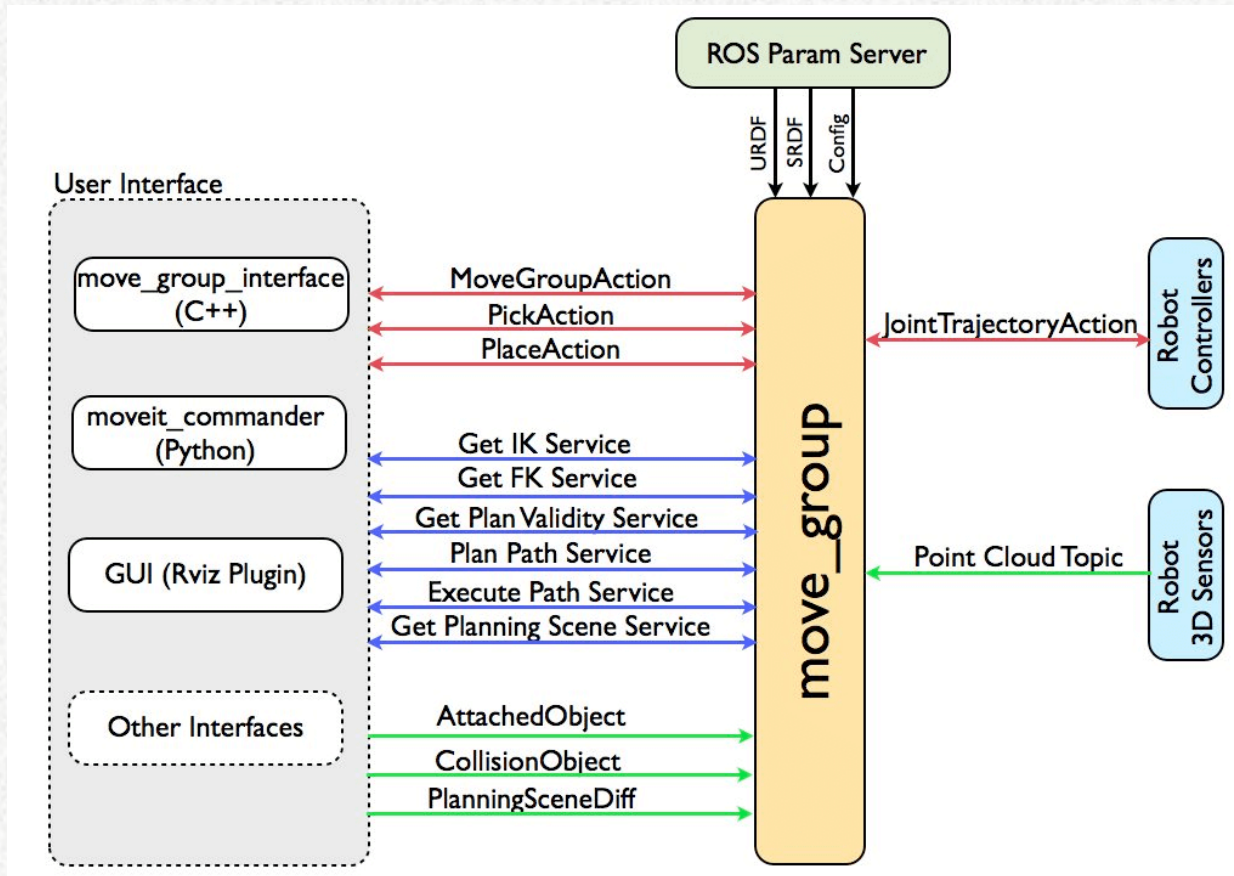
User Interface

- A `move_group` által biztosított service-ek és action-ök elérhetők több interface-en keresztül:
 - `move_group_interface` package (C++)
 - GUI – Rviz plugin-en keresztül
 - *`moveit_commander` package (Python) – nincs MoveIt2-ben*

Paraméterek

- A ROS paraméterként érhető el a MoveIt konfigurációs paraméterek (pl. csukló limit)
- Továbbá a robotleíró paraméterek is (`robot_description` - URDF, `robot_description_semantic` - SRDF)

Koncepció – move_group node



Robot Interface

- Kommunikáció a robottal: topic és action segítségével
 - Információk robottól `move_group` felé:
 - Aktuális robot állapot - `/joint_states` topic (pl. csukló pozíciók)
 - Transzformációk - `/tf` topic (pl. világ keret és robot keret között)
 - Pontfelhő
 - További robot szenzorok által mért adatok
 - Információk `move_group`-tól:
 - Robot kontrollereknek bemeneti adat (`FollowJointTrajectoryAction` interfész, ahol a robottal tartozik a szerver, a `move_group` a kliens)

Mozgástervezés MoveIt-tal

Motion Planning Plugin

- Mozgástervezők használata egy plugin interfészen keresztül
- Több könyvtár, különböző mozgástervezői használhatóak
- Könnyen bővíthető
- move_group node által szolgáltatott service vagy action a kommunikáció alapja
- Alapból rendelkezésre álló tervezők:
 - OMPL – Open Motion Planning Library (alapértelmezett tervezők)
 - randomizált tervezés – nem determinisztikus a megoldás
 - absztrakt tervezők – nem használják a robot modelljét.
 - Pilz Industrial Motion Planner
 - CHOMP – Covariant Hamiltonian Optimization for Motion Planning

Mozgástervezés MoveIt-tal

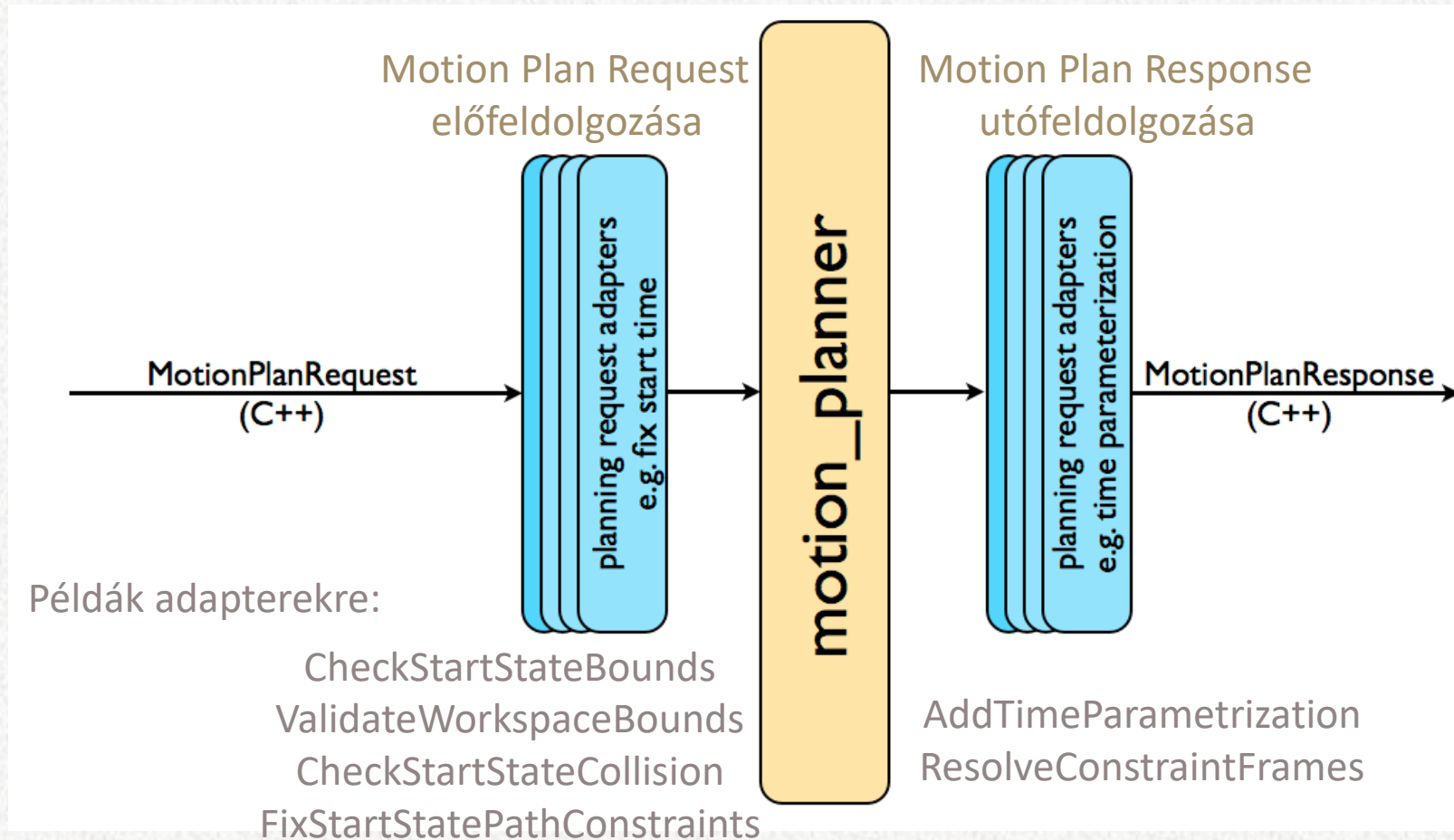
Motion Plan Request

- Megadja a mozgástervezési feladatot (pl. végberendezés mozgatása adott helyzetbe)
- Ütközések ellenőrzése (önütközés és megfogott objektum is)
- Beépített (és felhasználó által definiált) kinematikai kényszerek figyelembevétele:
 - Pozíció kényszerek (linkek vagy végberendezés pozíciójára)
 - Orientáció kényszerek (linkek vagy végberendezés orientációjára)
 - Láthatósági kényszerek (szenzor érzékelési területében belül maradó pontok)
 - Csukló kényszerek (limitek)

Motion Plan Result

- Egy trajektóriát határoz meg a move_group node a maximális sebességek és gyorsulások figyelembevételével. (Trajektória = pálya geometriája + sebességprofil)

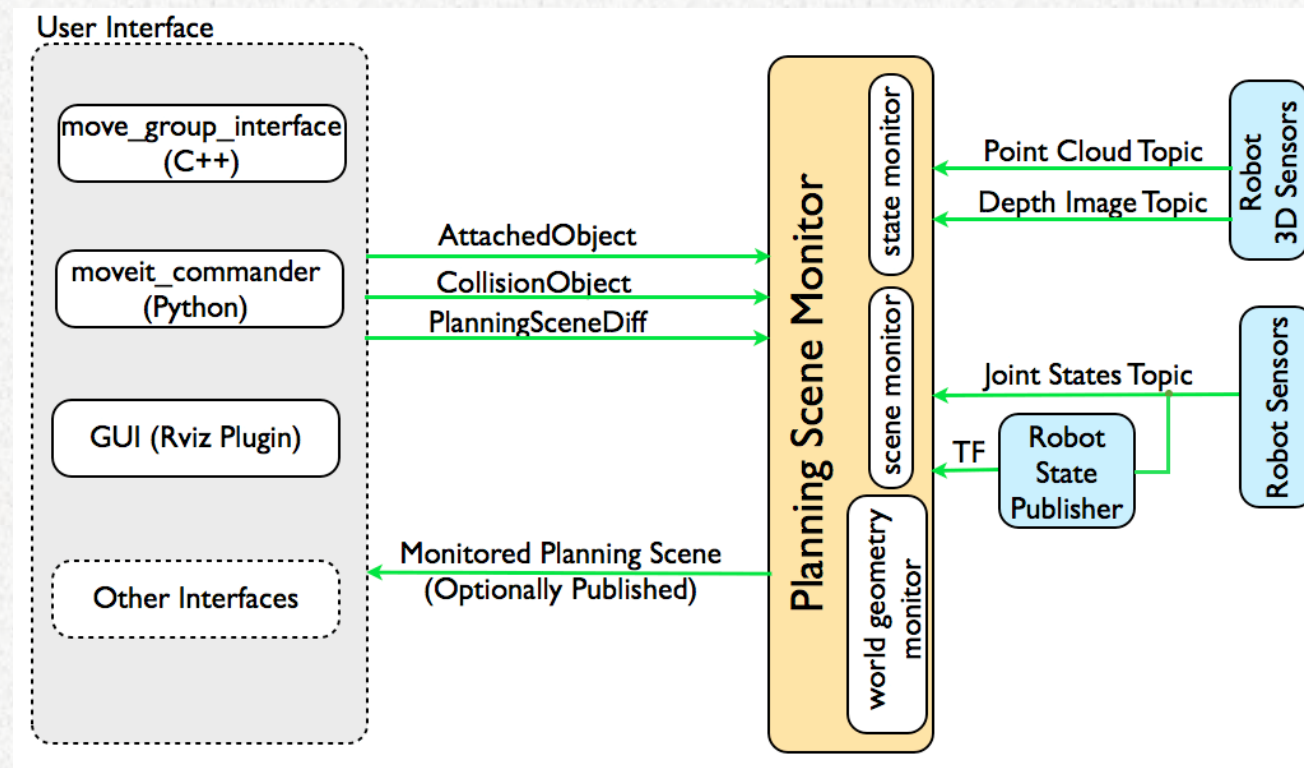
Motion Planning Pipeline



Planning Scene

Környezet és robotállapot reprezentáció:

- move_group node-ban levő Planning Scene Monitor az alábbi adatokat használja:
 - Állapot információ a /joint_states topic-ról
 - Szenzor információ (World Geometry Monitor használatával, 3D környezetretreprezentáció, foglaltsági térkép (occupancy map) pontfelhőből vagy mélységképekből)
 - Információ a környezetről (world geometry): /monitored_planning_scene topic-on megjelenő felhasználói változtatás



További funkciók

Kinematika

- Plugin, amivel akár saját inverz kinematika is implementálható ([IKFast package](#))

Ütközés detektálás

- Planning Scene-en belüli CollisionWorld objektum
- Általában FCL package végzi az ellenőrzést
- Ütközés vizsgálat végezhető a következő objektumokra
 - mesh
 - egyszerű alakzatok (téglatest, henger, gömb, stb.)
 - foglaltsági térkép ([OctoMap](#))
- ACM (Allowed Collision Matrix) – megadható, hogy melyik két objektum között nem kell elvégezni a vizsgálatot



Movelt eszközök

Movelt Setup Assistant

- Tutorial
- GUI - robot konfigurálása Movelt-hoz
- **SRDF** (Semantic Robot Description Format) generálás (szemantikus robot leírás)
 - URDF kiegészítése
 - csuklók csoportja definiálható
 - alap robot konfigurációk adhatók meg (pl. HOME pozíció)
 - további ütközésetektálási információk
 - további transzformációk
- konfigurációs fájlok előállítása (a robot Movelt config package-nek megfelelő config könyvtárba)
- launch fájlokat generál, amik a move_group plugin-ek konfigurációs paramétereit beállítják.

```

<group name="panda_arm">
  <joint name="virtual_joint"/>
  <joint name="panda_joint1"/>
  <joint name="panda_joint2"/>
  <joint name="panda_joint3"/>
  <joint name="panda_joint4"/>
  <joint name="panda_joint5"/>
  <joint name="panda_joint6"/>
  <joint name="panda_joint7"/>
  <joint name="panda_joint8"/>
</group>
<group name="hand">
  <link name="panda_hand"/>
  <link name="panda_leftfinger"/>
  <link name="panda_rightfinger"/>
</group>

```

```

<group_state name="ready" group="panda_arm">
  <joint name="panda_joint1" value="0"/>
  <joint name="panda_joint2" value="-0.7854"/>
  <joint name="panda_joint3" value="0"/>
  <joint name="panda_joint4" value="-2.3562"/>
  <joint name="panda_joint5" value="0"/>
  <joint name="panda_joint6" value="1.5708"/>
  <joint name="panda_joint7" value="0.7854"/>
</group_state>
<group_state name="extended" group="panda_arm">
  <joint name="panda_joint1" value="0"/>
  <joint name="panda_joint2" value="0"/>
  <joint name="panda_joint3" value="0"/>
  <joint name="panda_joint4" value="-0.1"/>
  <joint name="panda_joint5" value="0"/>
  <joint name="panda_joint6" value="3.1416"/>
  <joint name="panda_joint7" value="0.7854"/>
</group_state>

```

```

<disable_collisions link1="panda_link5_sc" link2="panda_rightfinger" reason="Default"/>
<disable_collisions link1="panda_link6" link2="panda_link6_sc" reason="Adjacent"/>
<disable_collisions link1="panda_link6" link2="panda_link7" reason="Adjacent"/>
<disable_collisions link1="panda_link6" link2="panda_link7_sc" reason="Always"/>
<disable_collisions link1="panda_link6" link2="panda_rightfinger" reason="Never"/>
<disable_collisions link1="panda_link6_sc" link2="panda_link7" reason="Always"/>

```

```

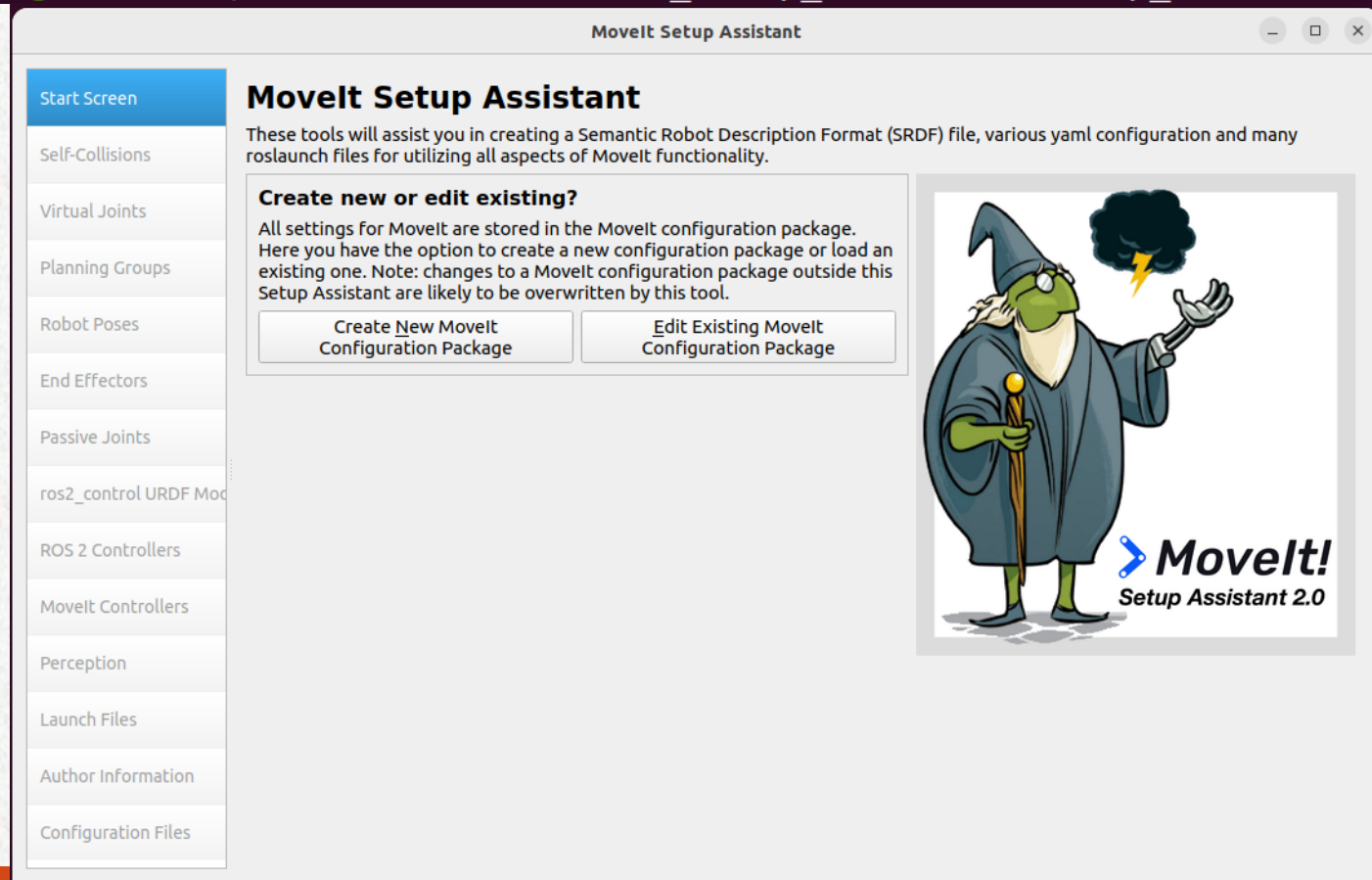
<virtual_joint name="virtual_joint" type="fixed" parent_frame="world" child_link="panda_link0"/>

```

Movelt Setup Assistant

Indítás:

```
emese@ubuntu:~$ ros2 launch moveit_setup_assistant setup_assistant.launch.py
```



Movelt Setup Assistant

Kihagyhatók az ütközés-vizsgálatból azok a szegmens párok, amik:

- sose ütköznek
- mindig ütköznek
- robot default pozíciójában ütköznek
- egymást követő szegmensek

Start

Self-Collisions

Virtual Joints

Planning Groups

Robot Poses

End Effectors

Passive Joints

Controllers

Simulation

3D Perception

Author Information

Configuration Files

Optimize Self-Collision Checking

This searches for pairs of robot links that can safely be disabled from collision checking, decreasing motion planning time. These pairs are disabled when they are always in collision, never in collision, in collision in the robot's default position, or when the links are adjacent to each other on the kinematic chain. Sampling density specifies how many random robot positions to check for self collision.

Sampling Density: Low High 10000

Min. collisions for "always"-colliding pairs: 95%

	panda_link0	panda_link1	panda_link2	panda_link3	panda_link4	panda_link5	panda_link6	panda_link7	panda_hand	panda_leftfinger
panda_hand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	<input type="checkbox"/>	✓	✓		✓
panda_leftfinger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	<input type="checkbox"/>	✓	✓	✓	
panda_rightfinger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	<input type="checkbox"/>	✓	✓	✓	<input type="checkbox"/>
panda_hand_sc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	✓	✓	✓	✓	✓
panda_link7_sc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	✓	✓	✓	✓	✓
panda_link6_sc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	✓	✓	✓	✓	✓
panda_link5_sc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	✓	✓	<input type="checkbox"/>	<input type="checkbox"/>	✓	<input type="checkbox"/>

link name filter

☐ linear view ☒ matrix view

☒ visual ☐ collision

Movelt Setup Assistant

Start
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
Controllers
Simulation
3D Perception
Author Information
Configuration Files

- **Virtuális csukló:** Ez köti hozzá a robotot a környezetéhez
- **Tervezési csoportok:** robotot különböző részei (pl. kar, végberendezés) szerint csoportokra lehet szedni
- **Robot pózok:** nevesített, gyakran használt konfigurációk (pl. home) definiálása a csuklópozíciók megadásával
- **Végberendezés:** ehhez a csoporthoz speciális műveletek tartoznak
- **Passzív csukló:** nem irányított, beavatkozó nélküli csukló (pl. mobilis robot bolygókeréke)
- **Controllers:** valós robot irányítása
- **Szimuláció:** URDF módosítása Gazebo szimulációhoz
- **3D érzékelés:** szenzornak megfelelő paraméterek megadása

További MoveIt és MoveIt2 eszközök

- [HandEyeCalibration](#): GUI kamera kalibrációhoz
- [MoveIt Grasps](#): megfogás generálás a végberendezéssel
- [MoveIt Task Constructor \(MTC\)](#): olyan összetett műveletek tervezése, amik több részfeladatból állnak.
- [Gazebo Simulation Integration](#): Megfelelő módosítások után a robot szimulálható Gazebo-ban
- [Benchmarking](#): Mozgástervezők értékelése, összehasonlítása
- [Hybrid Planning](#): Lokális és globális mozgástervezők alkalmazása változó környezetben
- Tervezés [mobil manipulátor](#) számára
- És még van pár: [MoveIt tutorials](#), [MoveIt2 tutorials](#)



Movelt - programozás

Programozás

- Kódból is végezhetők hasonló műveletek, mint az RViz pluginnal:
 - csuklópozíciók megadása, cél beállítása, mozgástervezés, robot mozgatás, objektum hozzáadása a környezethez, objektum robohoz csatolása és lecsatolása
- Egy példa MoveIt projekt C++-ban
- Move Group C++ Interface
- C++ API használata
- Python API használata