

2015/8/4

# RTMサマーキャンプ2015

## DARPA Robotics Challenge における ChoreonoidとRT-Middlewareの活用

産業技術総合研究所

知能システム研究部門ヒューマノイド研究グループ

中岡 慎一郎

# 産総研で開発されたヒューマノイドロボット



HRP-2P  
(2002)

HRP-2  
(2002)

HRP-3P  
(2005)

HRP-3  
(2007)

HRP-4C  
(2009)

HRP-4  
(2010)

HRP-2改  
(2015)



# DARPA Robotics Challenge (DRC)





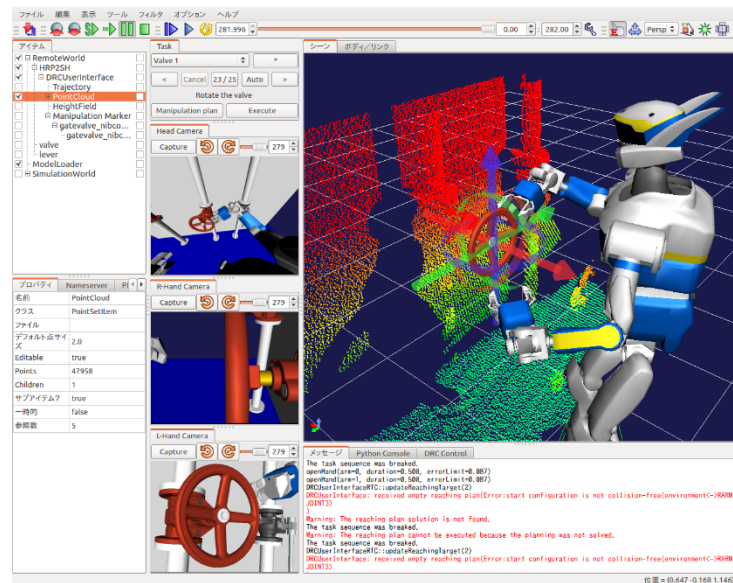
17:46:06 06/06/2015

# 遠隔操作システム

オペレータ



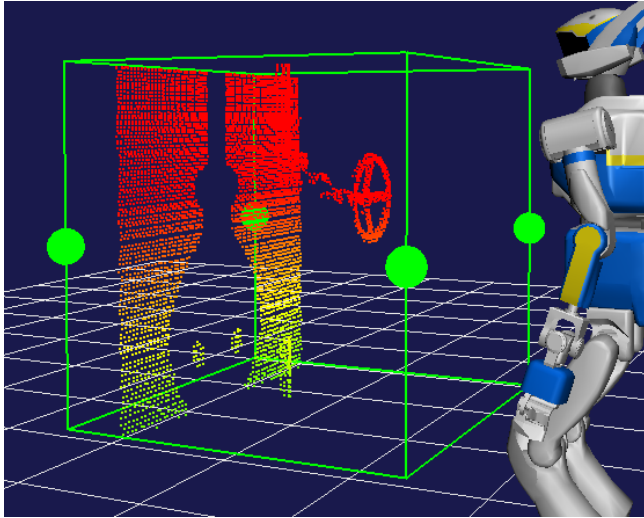
GUI (Choreonoid)



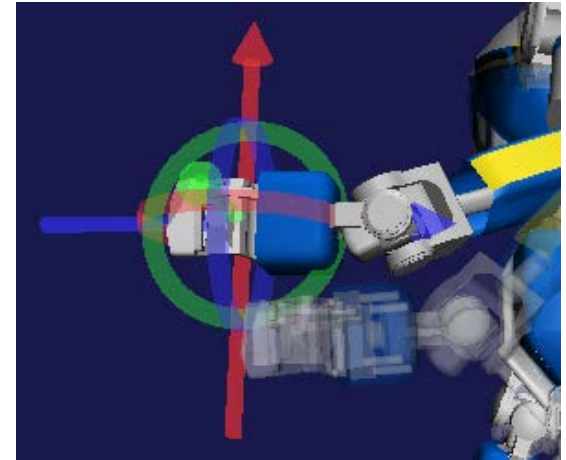
実ロボット



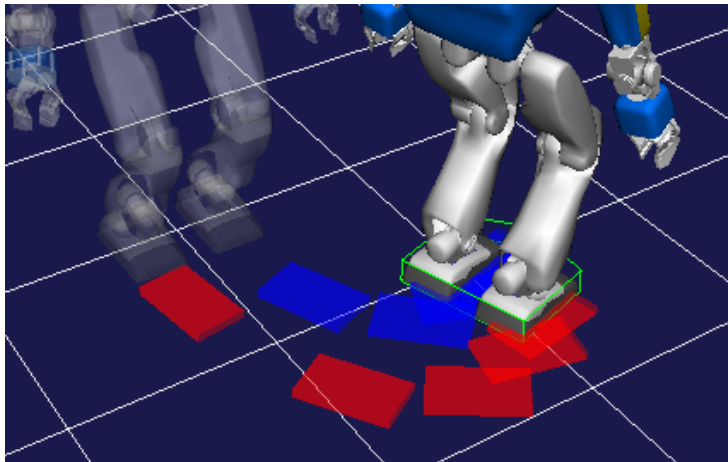
# 操作用マーカー



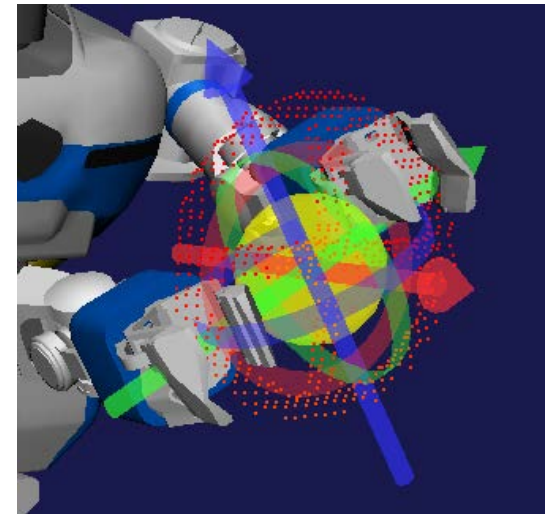
計測領域指定マーカー



手先目標点指定マーカー



歩行先指定／フットステップ表示マーカー



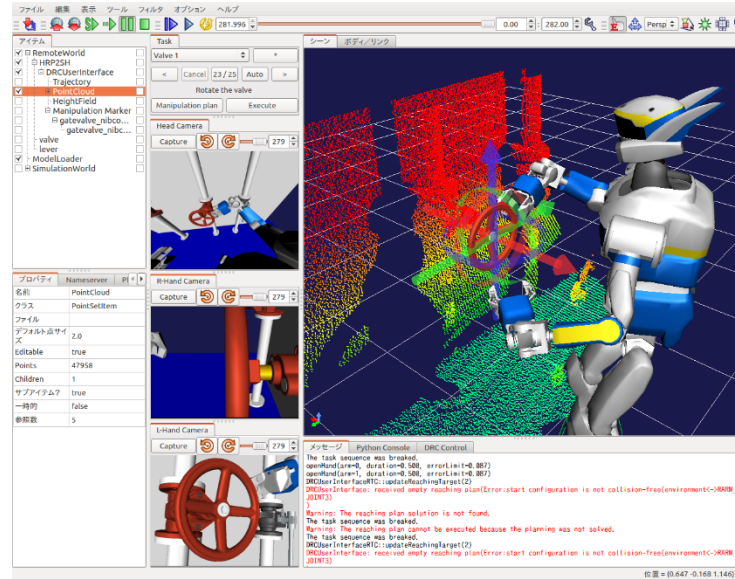
物体操作マーカー

# 遠隔操作システム

オペレータ



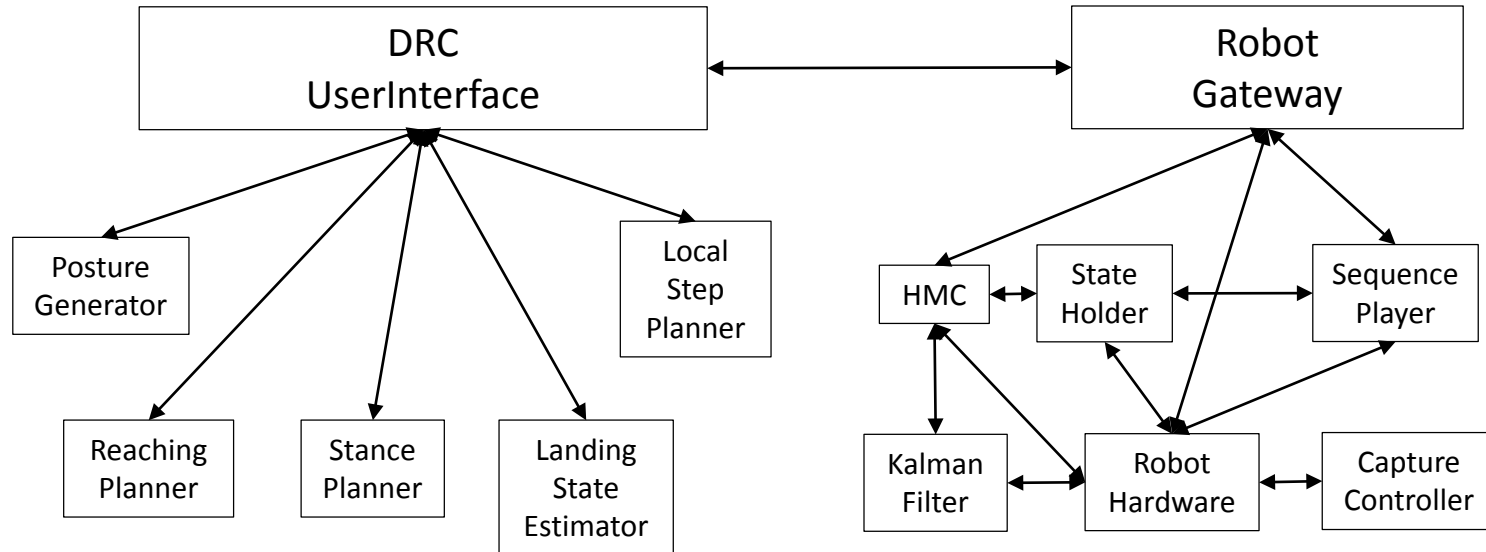
GUI (Choreonoid)



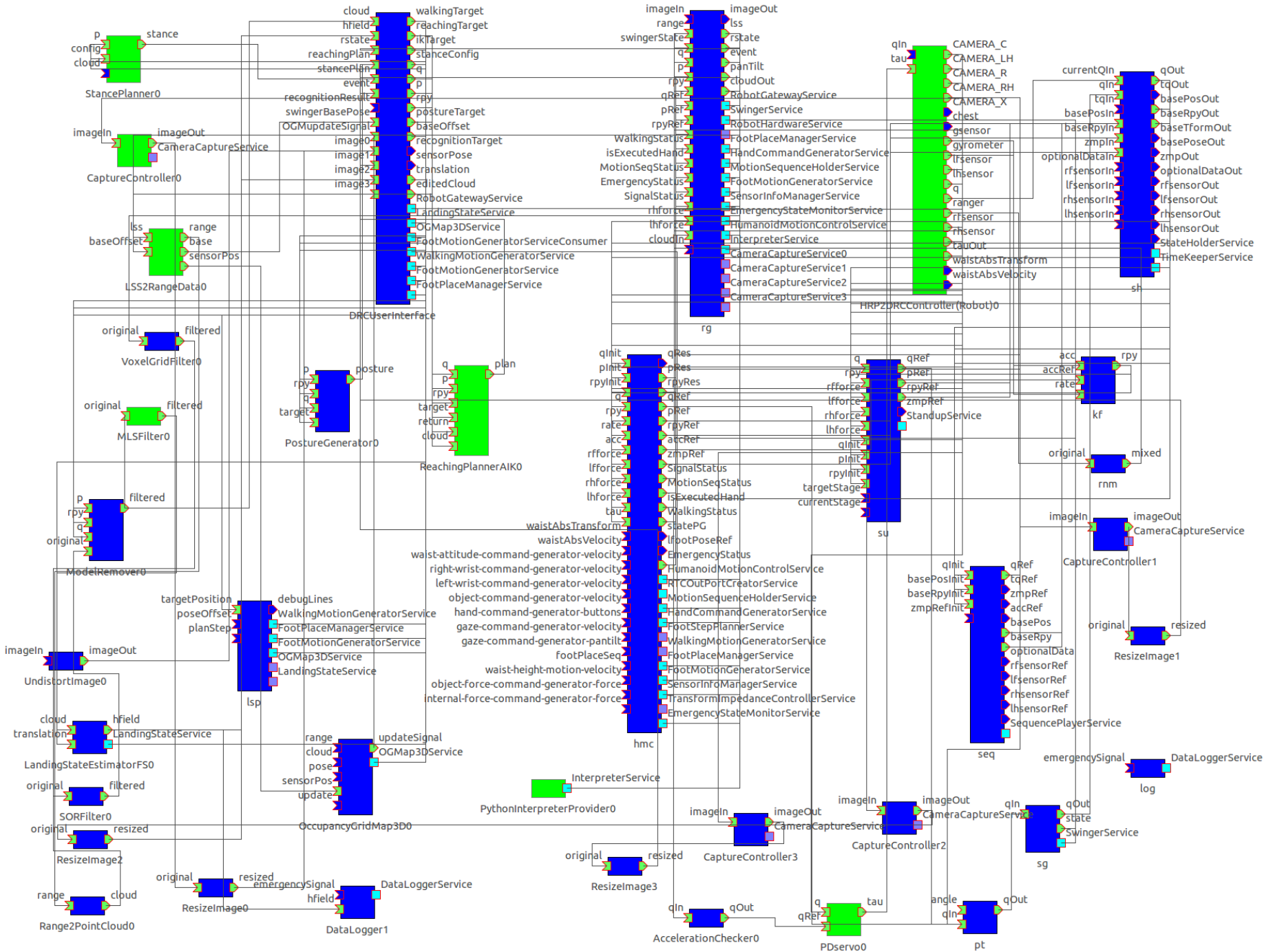
実ロボット



RT Components





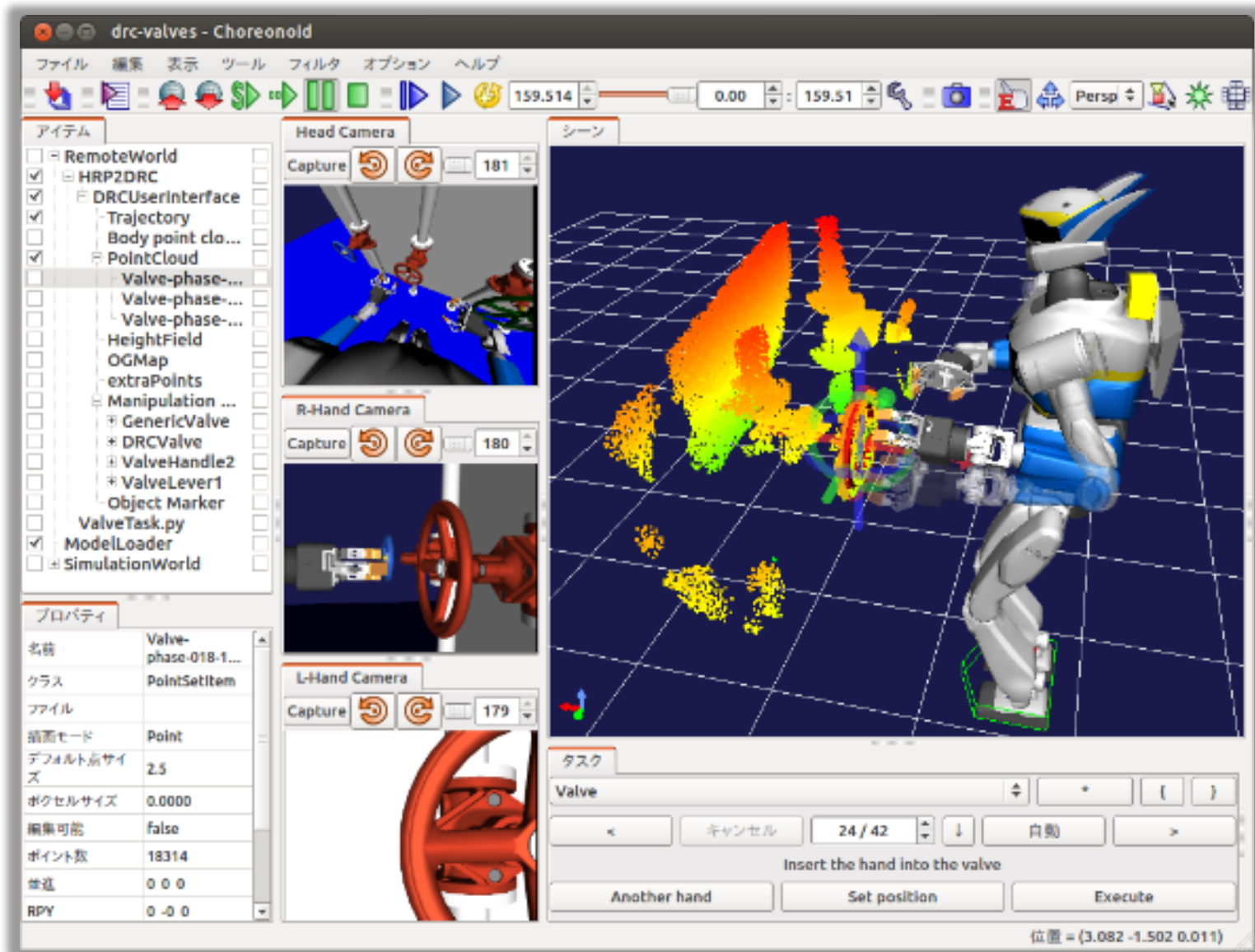


# PythonスクリプトによるRTCの生成・接続

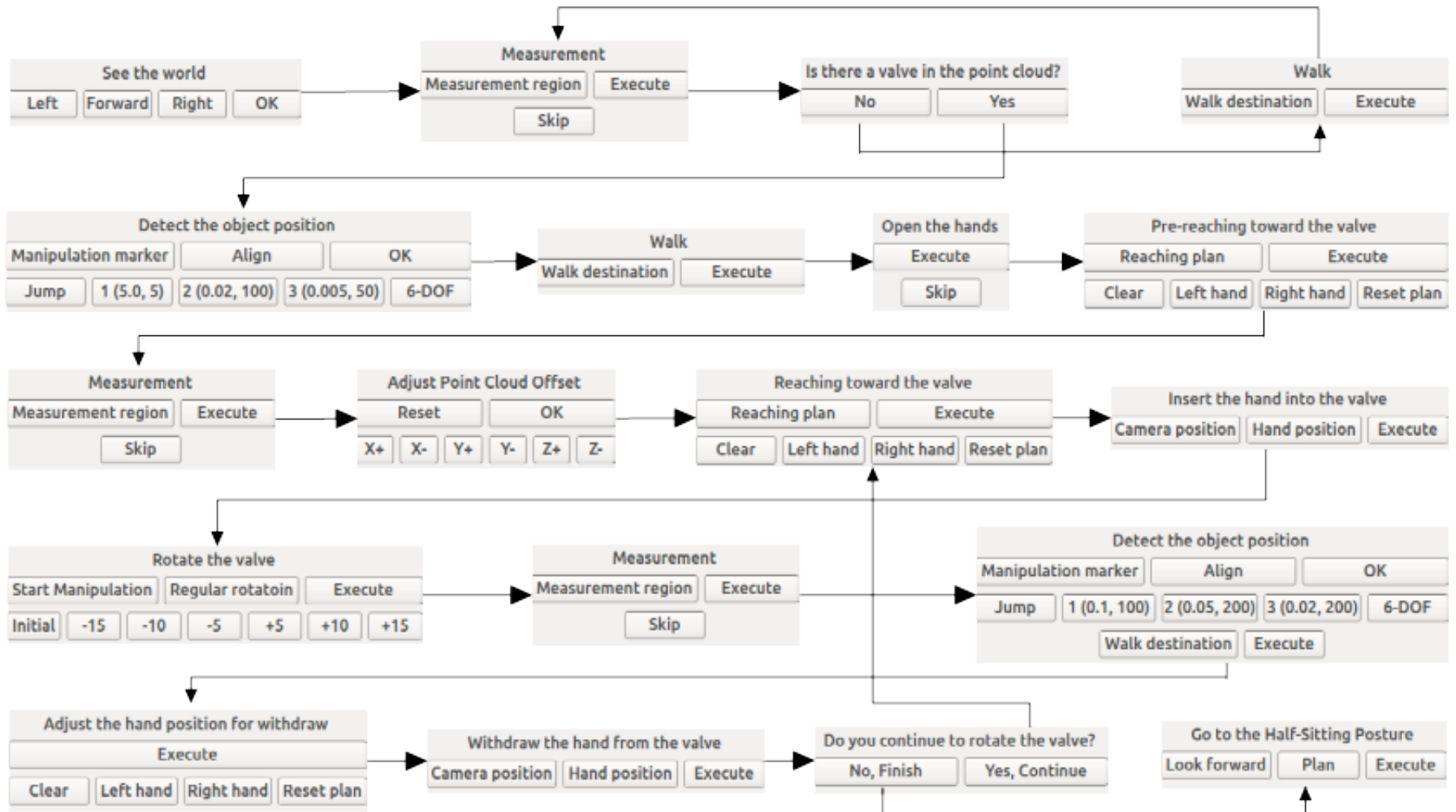
```
def createComps():
    ms = rtm.findRTCmanager()
    rh = rtm.findRTC("RobotHardware0")
    ms.load("SequencePlayer")
    seq = ms.create("SequencePlayer", "seq")
    seq_svc = narrow(seq.service("service0"), "SequencePlayerService")
    ms.load("KalmanFilter")
    kf = ms.create("KalmanFilter", "kf")
    :
    :

def connectComps():
    connectPorts(rh.port("q"), [sh.port("currentQIn"), hmc.port("q")])
    connectPorts(rh.port("rfsensor"), hmc.port("rfforce"))
    connectPorts(rh.port("lfsensor"), hmc.port("lfforce"))
    connectPorts(rh.port("rhsensor"), hmc.port("rhforce"))
    connectPorts(rh.port("lhsensor"), hmc.port("lhforce"))
    connectPorts(rh.port("gyrometer"), [kf.port("rate"), hmc.port("rate")])
    connectPorts(rh.port("gsensor"), [kf.port("acc"), hmc.port("acc")])
    :
    :
```

# 遠隔操作デモ



# タスクシーケンス



# 攻略タスク(5ポイント)



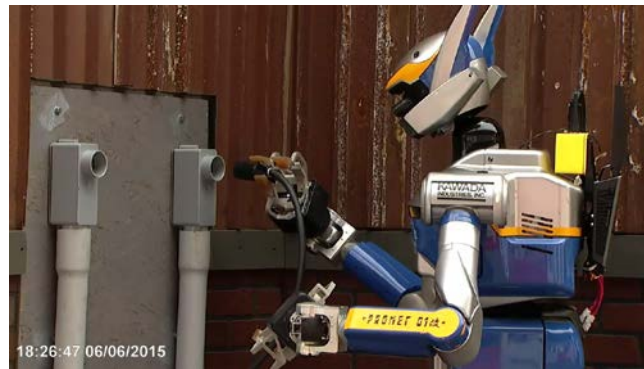
1. 運転



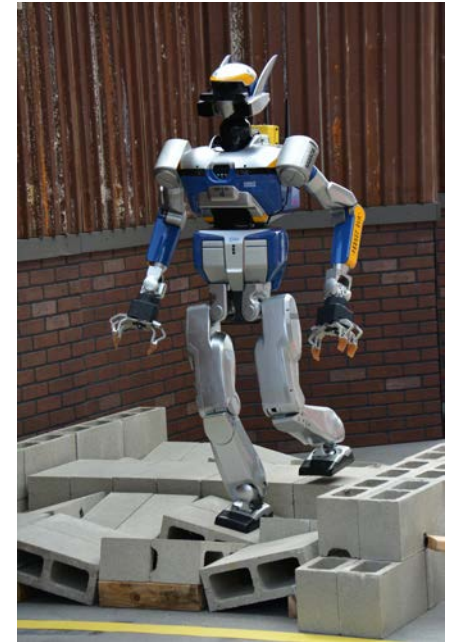
2. ドア



3. バルブ



4. サプライズ(プラグ)



5. 不整地  
(クリアと同時に転倒)

# 成績

KAIST  
DRC-HUBO



IHMC  
Atlas



Tartan  
CHIMP



NimbRo  
Momaro



RoboSimian



Position	TEAM	SCORE	TIME
1	KAIST	8	44:28
2	IHMC ROBOTICS	8	50:26
3	TARTAN RESCUE	8	55:15
4	NIMBRO RESCUE	7	34:00
5	ROBOSIMIAN	7	47:59
6	MIT	7	50:25
7	WPI-CMU	7	56:06
8	DRC-HUBO AT UNLV	6	57:41
9	TRAC LABS	5	49:00
10	AIST-NEDO	5	52:30
11	NEDO-JSK	4	58:39

# まとめ

- OpenRTMでこの規模のロボットシステムもやってみたらちゃんと作れた
- コンポーネントが多くなると、GUIツールでシステムを構成するのには無理がある
- Pythonスクリプトの併用が有効
- DRCでは多くのチームがROSを使用
- OpenRTMは、制御ループ(リアルタイム)にも対応可能なところがウリ？