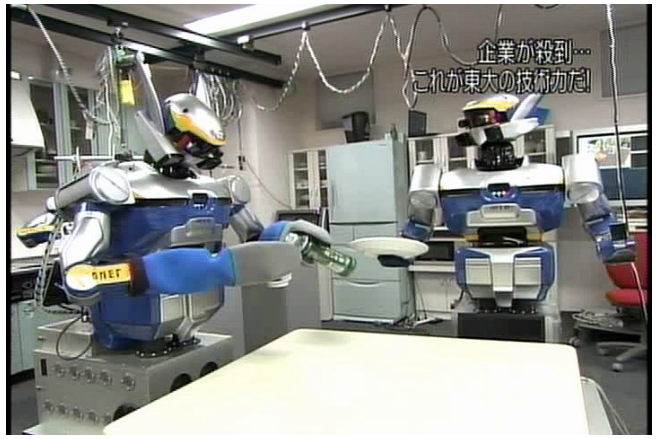
Building Interoperability between OpenRTM and ROS

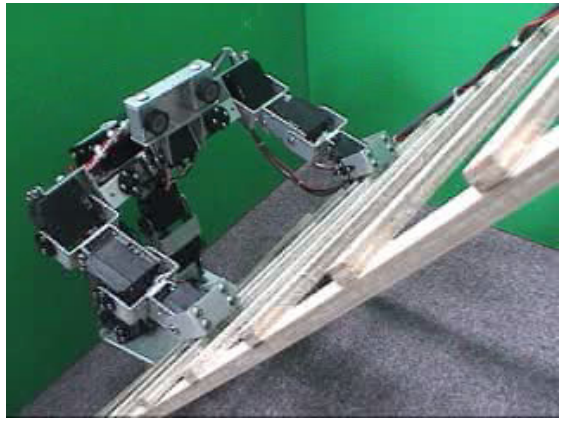
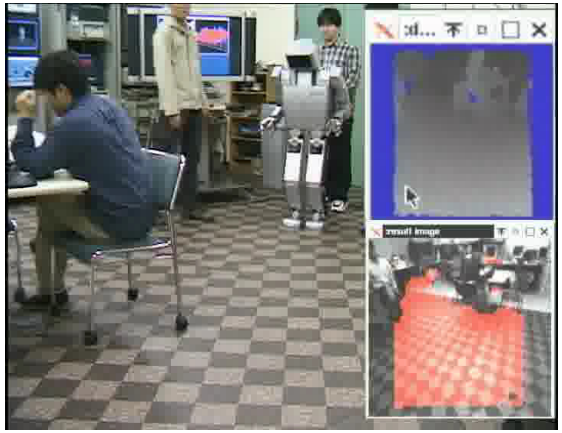
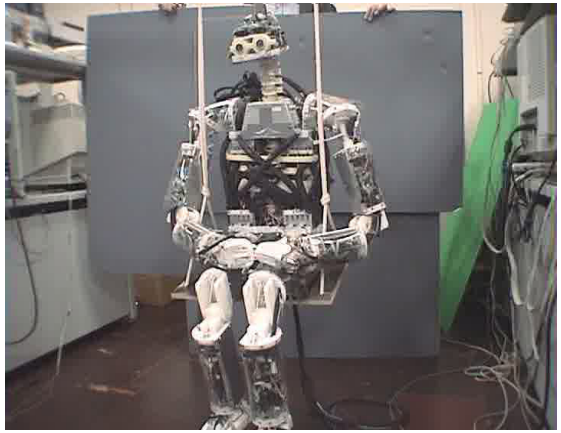
The University of Tokyo / AIST
Kei Okada
k-okada@jsk.t.u-tokyo.ac.jp



Outline

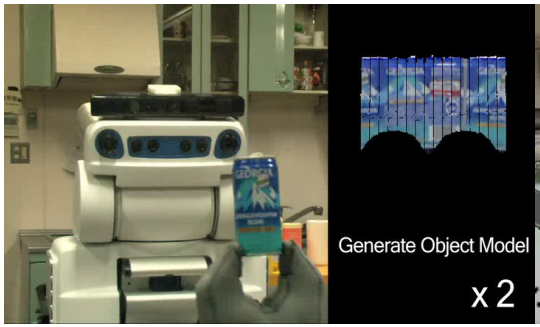
- Introduction
- OpenRTM and ROS Integration
- Using OpenRTM component for ROS users
- Using ROS node for OpenRTM users
- Conclusion

Robots in JSK Lab (2002)

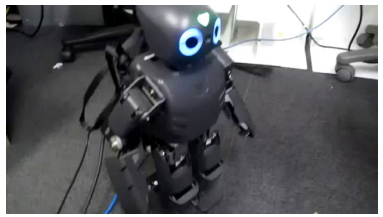
	Remote-brained Robot	Life-sized Humanoid	Musculo-skeletal Humanoid
			
hard ware	Custom made ware	Custom made hardware	Custom made hardware
comp uter	Micro computer + PC	Embedded PC	Distributed MPU + PC
Middle-ware	custom made framework (nervous: plugin-component framework for humanoid)	custom made framework (RT-Linux kernel module → nervous)	custom made framework (nervous: plugin-component framework for humanoid)
Top-level	EusLisp robot programming environment		

→ Towards uniform robotics software framework

Robots in JSK Lab (2012)

Mobile Manipulator	Life-sized Humanoid	Musculo-skeletal Humanoid	Dynamic Humanoid
			
WillowGarage	Kawada Inc.	Custom made	Custom made
PC Servers	Embedded PC	Distributed MPU + PC	Embed. PC
ROS nodes	RTM components on OpenHRP3 and hrpsys	Custom made framework (nervous)	Custom made
EusLisp robot programming environment			

Humanoid



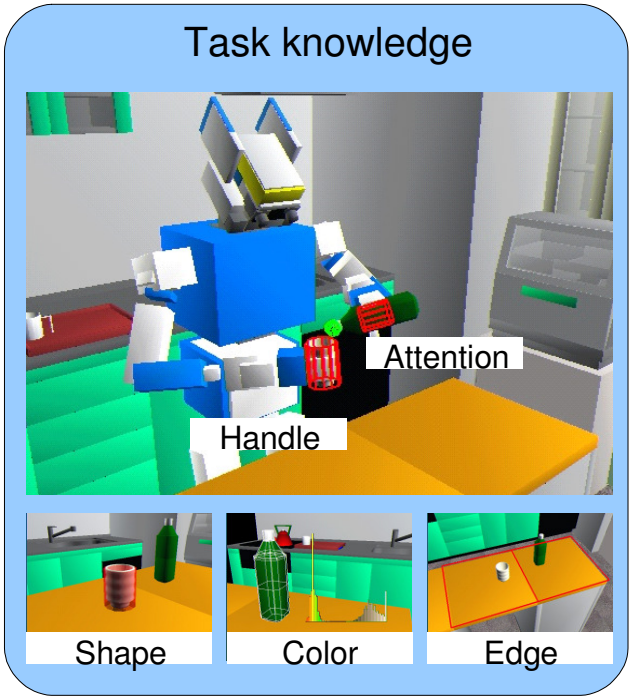
ROBOTIS

Embed. PC

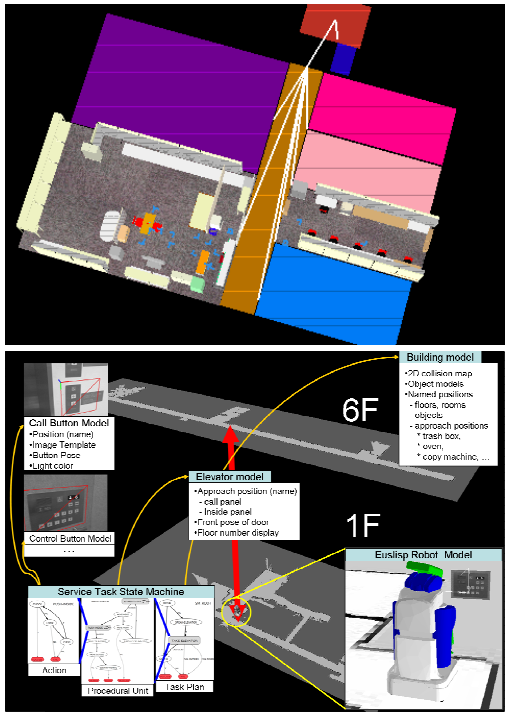
DarwIn-OP

EusLisp programming environment

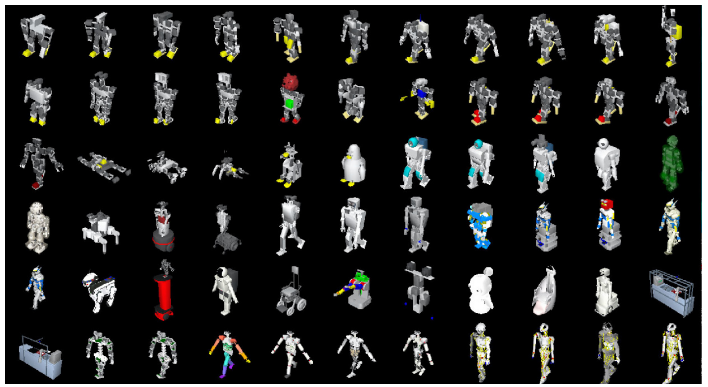
- 1986, Toshihiro Matsui@AIST
- Interpreter with solid modeling functions
- Modeling robots and objects
  + scene or task knowledge



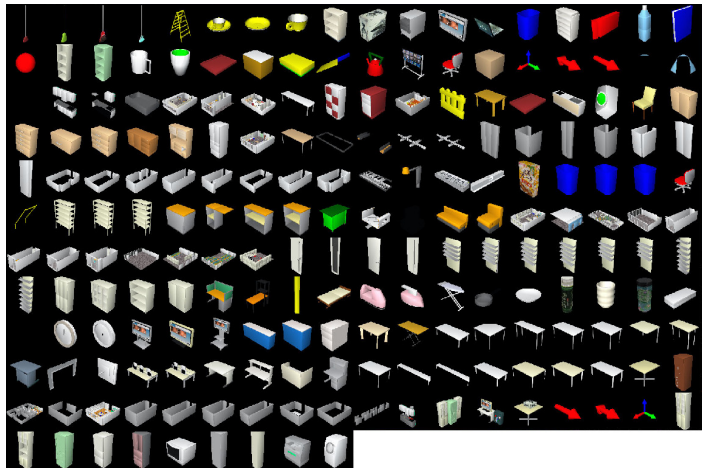
Task related knowledge



Scene knowledge



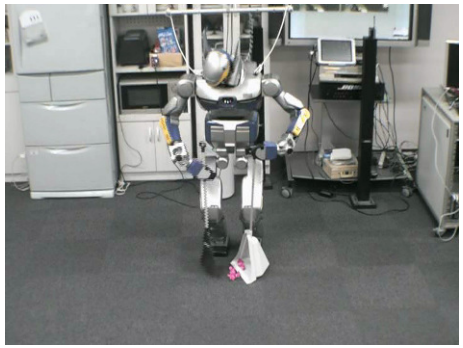
Robot models

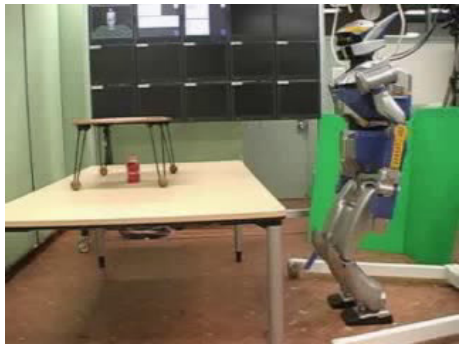


Object models

Daily assistive tasks of HRP2JSK

HRP2JSK(2002-)
Tasks generated based on humanoid motion planning technology





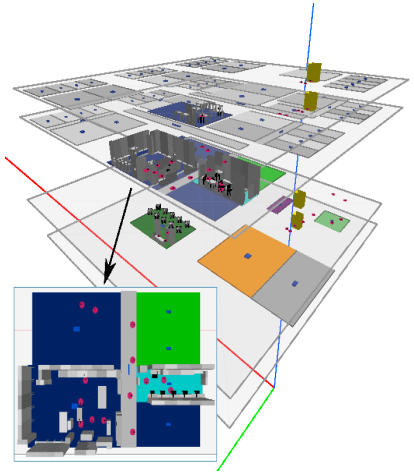
Tool manipulation

Clean up

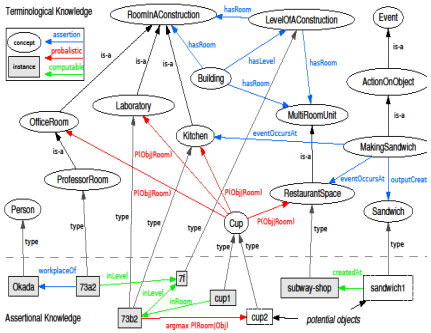
In the kitchen

Play ;-)

ROS based framework for integrate JSK and IAS software

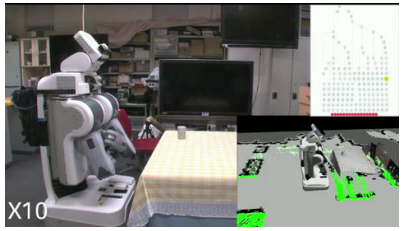


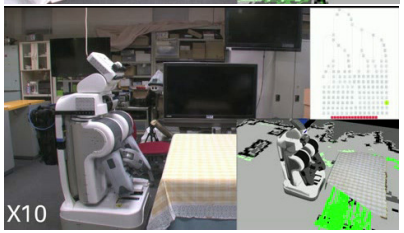
IAS semantic map

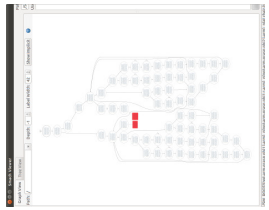


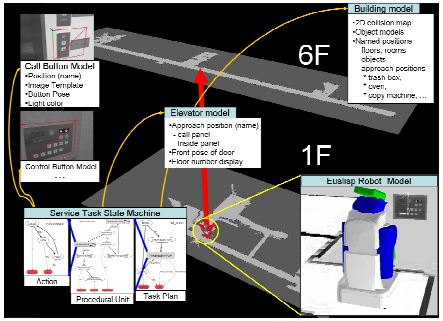
IAS knowrob map

- Inferring likely location for a sandwich using web based semantic environment models

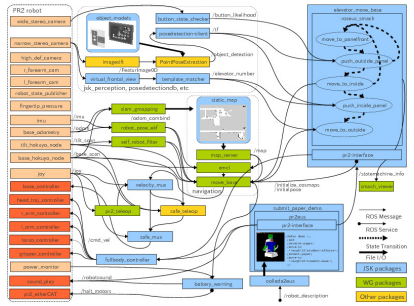








JSK task knowledge

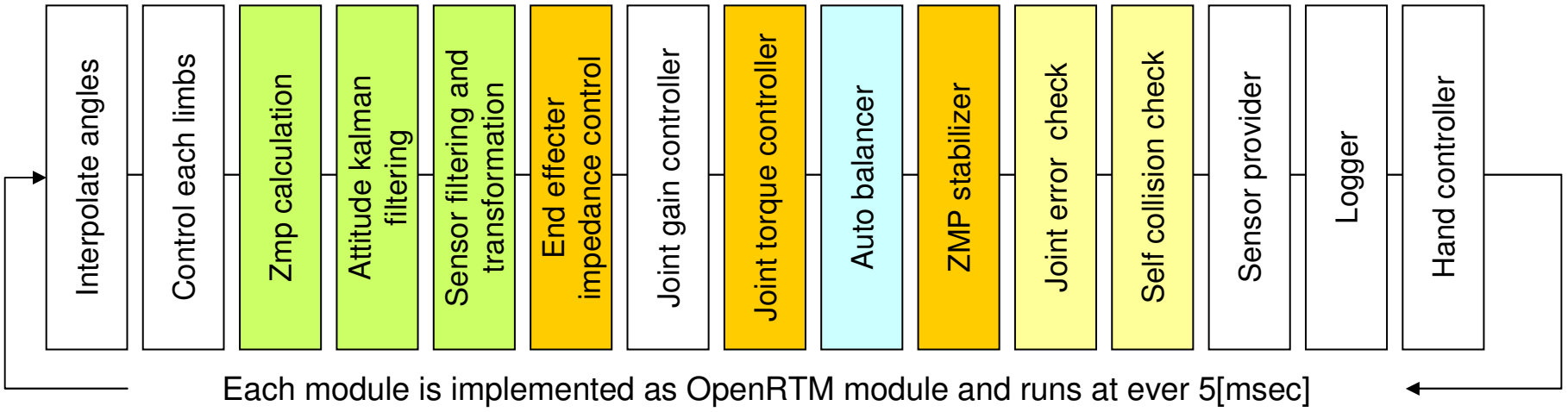


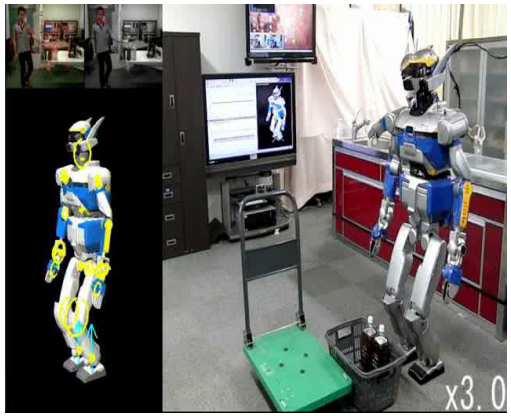
ROS nodes conn.

← Task planner to generate behavior executives

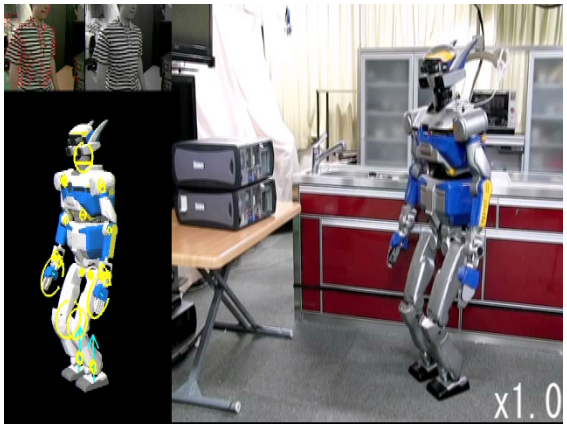
OpenRTM components on HRP Humanoids

Feed forward motion
Sensor feedback motion
Interface
Fail-safe softwares
Sensor data processing





Force sensor monitoring
(StateProvider, StateProvider2
ZMPsensor,Torquecontrol)



CoG modification based on Force sensor heedback
(AutoBalancer)

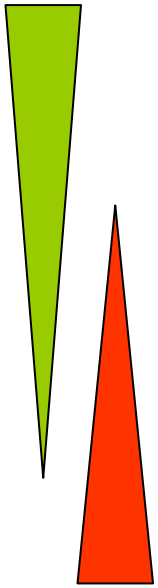
Fullbody motion generation
Based on Joint Torque control
(Torque Control)

OpenRTM and ROS integration

What is ROS exactly?

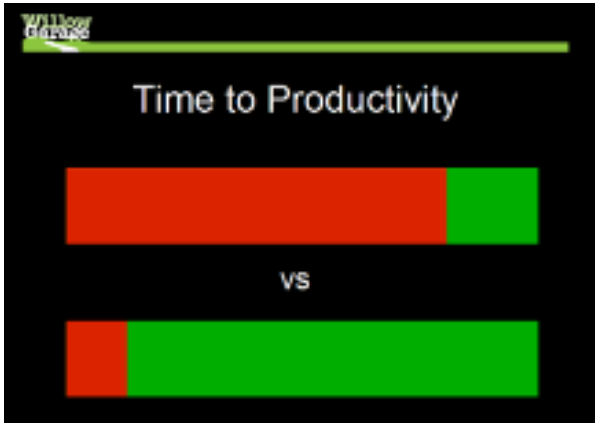
ROS = plumbing + tools + capabilities + ecosystem
B. Gerkey, Dec 06 '11. answers.ros.org

- Application
- Modules
- Library
- Simulator
- Communication
- Device Drivers
- Tools



Research
Target of OpenRTM project

Tools
ROS provides extensive set in this layer



Red indicates time to build tools,and greer shows the research. Current PhD student spend most of their time to build tools. ROS is designed to provide efficient tools for researchers to concentrate on the "research"（Steve Cousins speaking at Robo Development：
http://www.willowgarage.com/blog/2008/11/17/ste ve-cousins-speaking-robo-development-tuesday)

→ Builing OpenRTM-ROS environment on ROS-tools

- Connecting OpenRTM and modules developed in all over the word.
- Efficient development and maintenance

http://rtm-ros-robotics.googlecode.com

1. Compile OpenRTM component with `rosmake`
   - rtmbulid/cmake/rtmbuild.cmake
   - Compile and link OpenRTM idl files
2. Invoke OpenRTM component with `roslaunch`
   - openrtm/scripts/rtmlaunch.py
   - Add `<rtconnect>` and `<rtactivate>` tag to support component management in launch file.

CMakeLists.txt example (hrpsys/CMakeLists.txt)

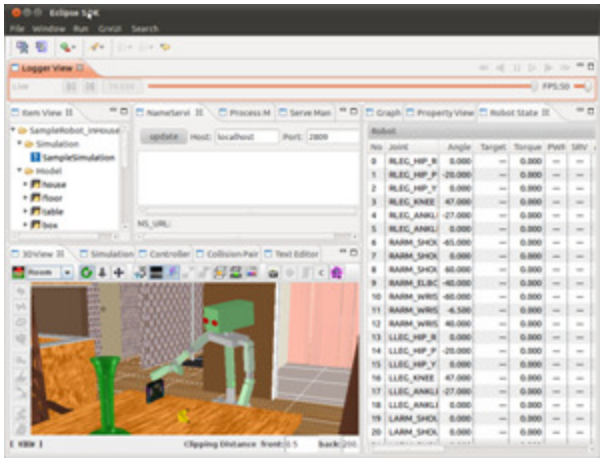
```
cmake_minimum_required(VERSION 2.4.6)
include($ENV{ROS_ROOT}/core/rosbuild/rosbuild.cmake)
rosbuild_find_ros_package(rtmbuild)
include(${rtmbuild_PACKAGE_PATH}/cmake/rtmbuild.cmake) # include rtmbuild.cmake
rtmbuild_init() # this calls rosbuild_init() inside
rtmbiild_genidl()                                      # compile idl
rtmbuild_add_executable(AbsTransformToPosRpy src/AbsTransformToPosRpy.cpp
    src/AbsTransformToPosRpyComp.cpp)                  # compile component
```

Launch file example (openrtm/launch/MobileRobot_2DSimulator_Example.launch)

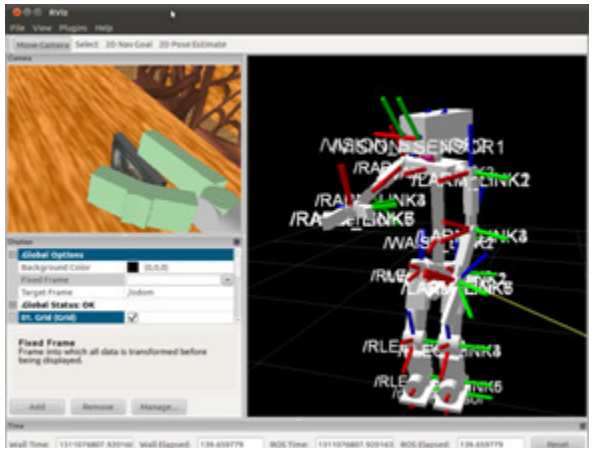
```
<node name="rtmlaunch" pkg="openrtm" type="rtmlaunch.py" args="$(find
    openrtm)/launch/MobileRobot_2DSimulator_Example.launch" />
<rtconnect from="TkJoyStick0.rtc:pos" to="TkMobileRobotSimulator0.rtc:vel" />
<rtactivate component="TkJoyStick0.rtc" />
<rtactivate component="TkMobileRobotSimulator0.rtc" />
```

OpenRTM-ROS Example

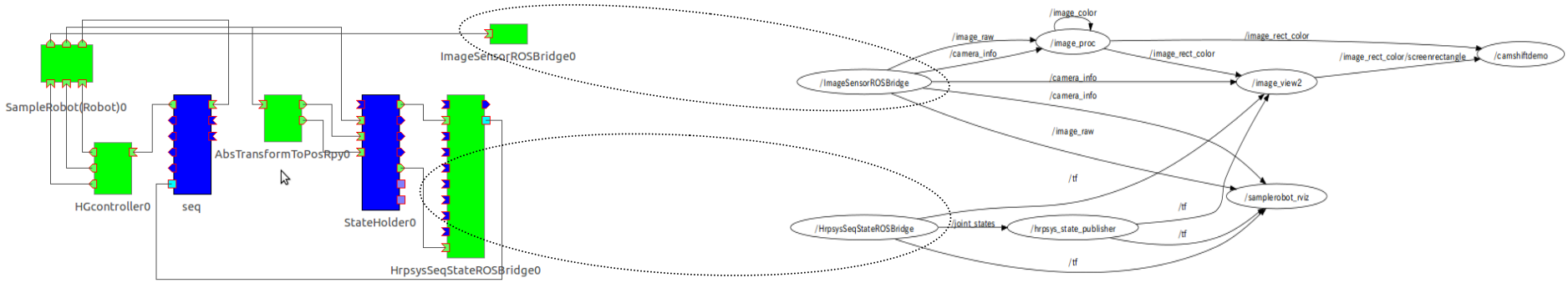
```
# Install RTMROS environment
$ rosinstall ~/prog/rtm-ros-robotics http://rtm-ros-robotics.googlecode.com/svn/tags/latest/agentsystem_ros_tutorials/rtm-ros-robotics.rosinstall
# Compile RTMROS environment
$ rosmake hrpsys_ros_bridge
# Launch RTMROS environment
$ roslaunch hrpsys_ros_brikdge samplerobot.launch
```



RTM
- Dynamics simulation
- Robot Model
- Controller
- Sequence
- Sensor Holder

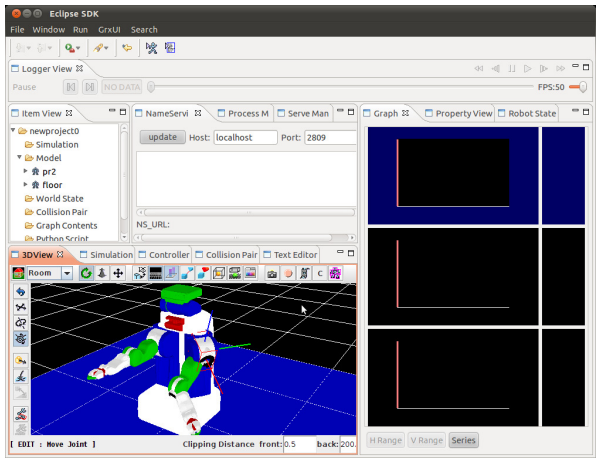


ROS
- Sensor Viewer
- Image Processin

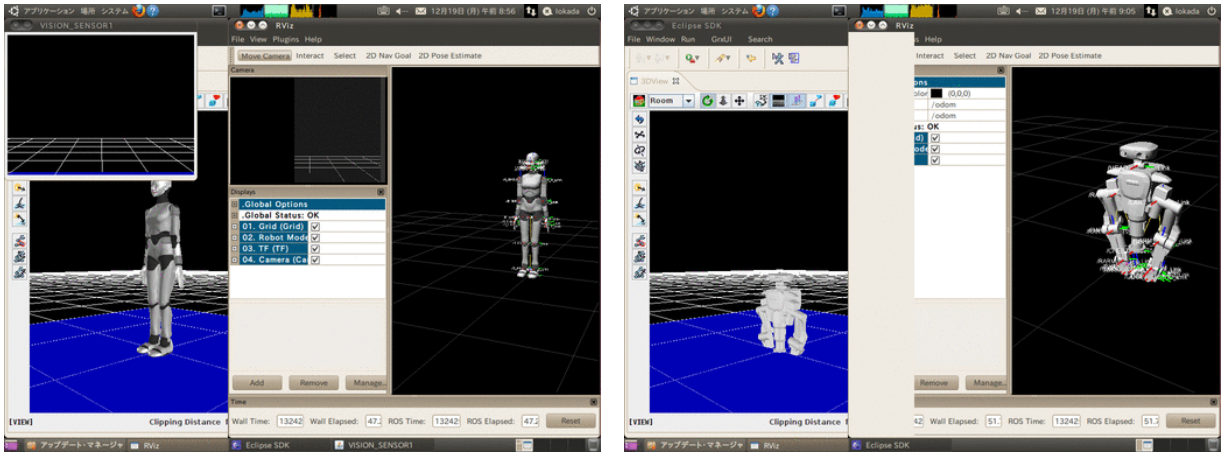


Interoperability of Robot Model between OpenRTM and ROS

- Collada support on both OpenRTM and ROS
  - Convert OpenRTM (OpenHRP3 wrl format) to Collada (openhrp3/export-collada)
  - Convert ROS (URDF format) to Collada (robot_model support collada natively)
- ROS package for OpenRTM software (OpenHRP3, hrpsys, Choreonoid, OpenVGR, OpenINVENT)



PR2 in OpenHRP3 simulator



Simulating in OpenHRP3 and display the robots in rviz

Using OpenRTM for ROS users

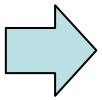
- `rtmbuild.cmake` provides following cmake macros

```
rtmbuild_init()
rtmbuild_genidl
rtmbuild_genbridge()
```

- Compile IDL file into ROS service messages file and OpenRTM/ROS source file

```
module SimpleService {
 typedef sequence<string> EchoList;
 typedef sequence<float> ValueList;
 interface MyService
 {
   string echo(in string msg);
   EchoList get_echo_history();
   void set_value(in float value);
   float get_value();
   ValueList get_value_history();
 };
};
```

SimpleService IDL File

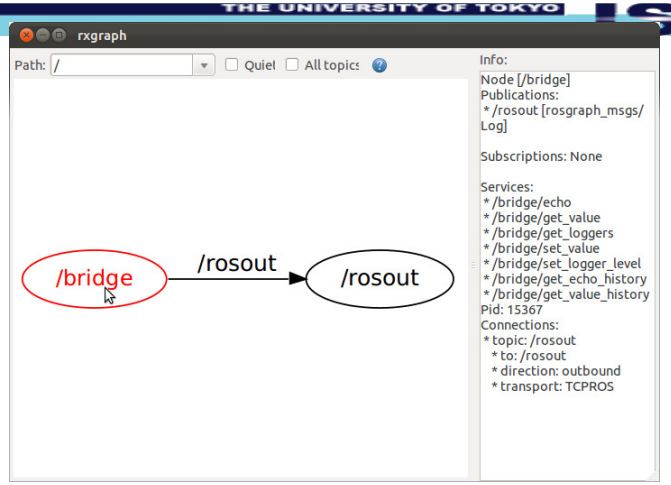


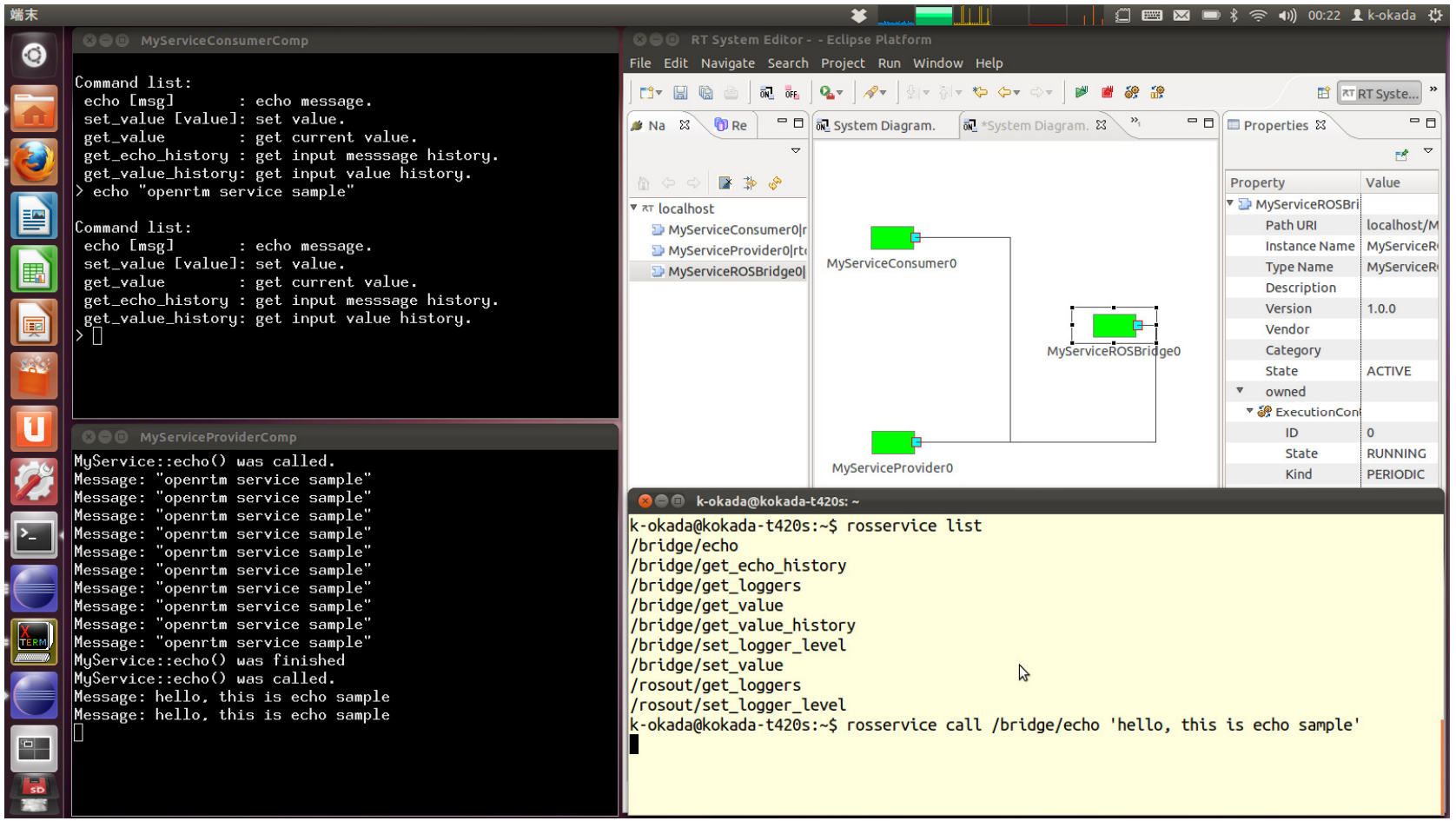
```
$ ls rc_gen/
MyServiceROSBridge.cpp
MyServiceROSBridge.h
MyServiceROSBridgeComp.cpp

$ ls srv
SimpleService_MyService_echo.srv
SimpleService_MyService_get_echo_history.srv
SimpleService_MyService_get_value.srv
SimpleService_MyService_get_value_hisotory.srv
SimpleService_MyService_set_value.srv
```

Generated Service messages

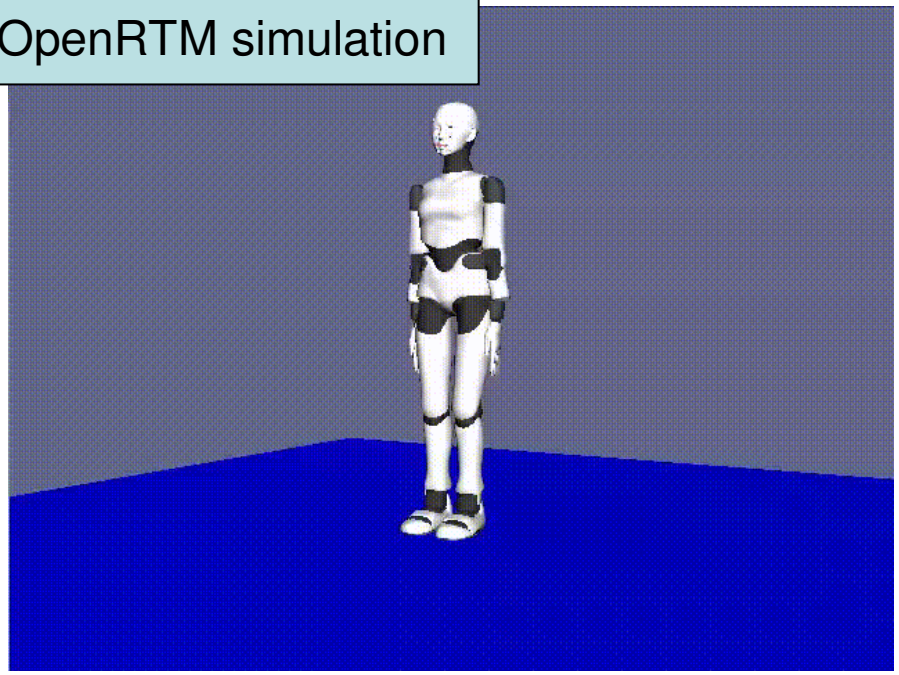
SimpleService example

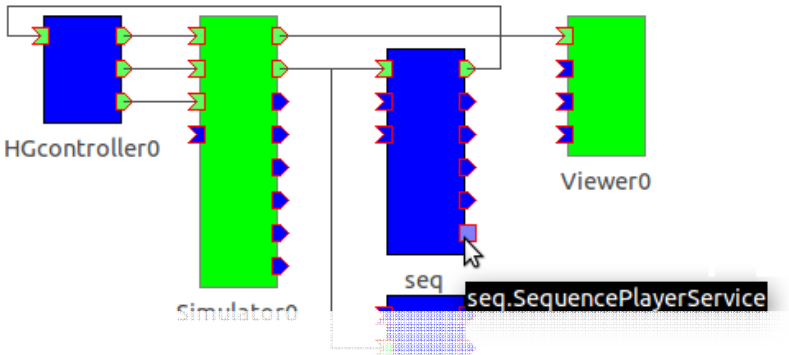
- Calling OpenRTM Simple Service example from rosservice



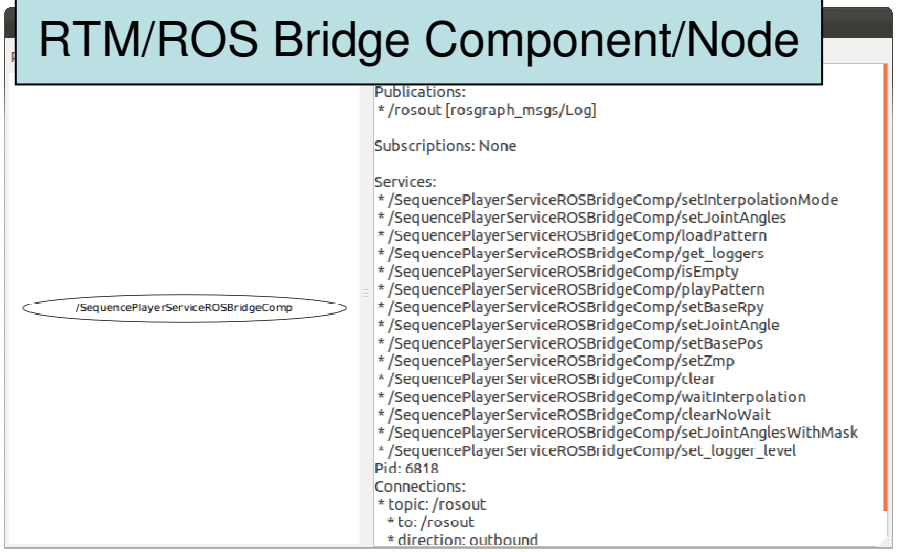


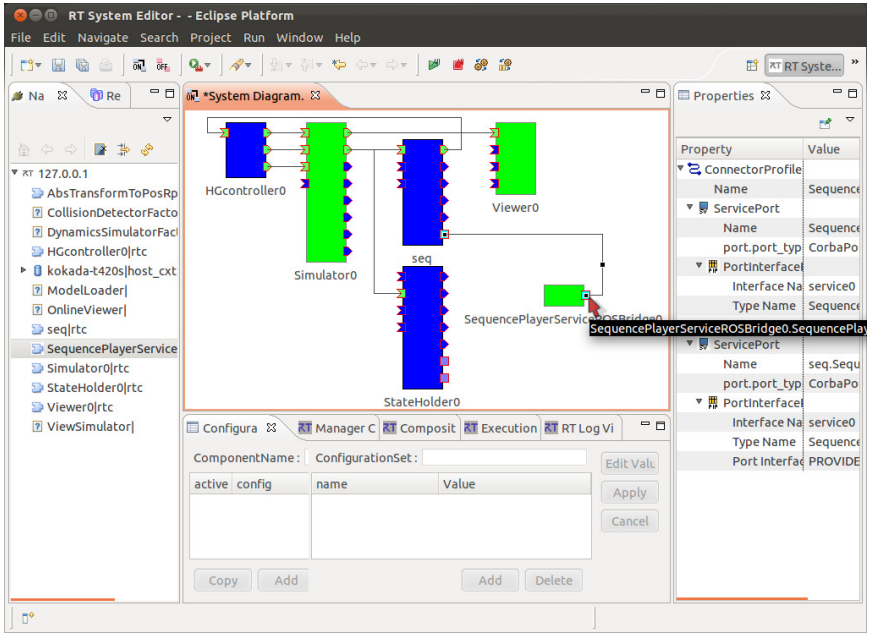
OpenRTM simulation





RTM/ROS Bridge Component/Node





Launch OpenRTM humanoid example

```
$ roslaunch hrpsys hrp4c.launch
$ rosrun openhrp3 rtsysedit.sh
```

Launch ROS-OpenRTM bridge component

```
$ roslaunch hrpsys hrp4c-rosbridge.launch
```

Example code to send walking pattern from rosservice command

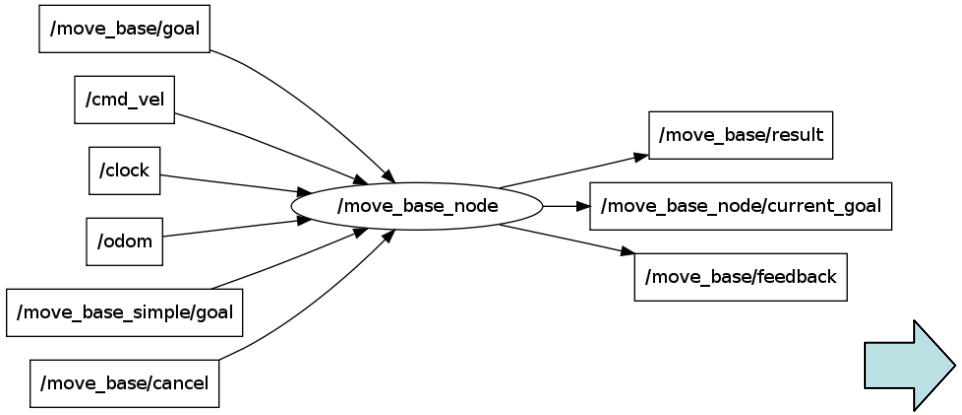
```
$ rosservice call /SequencePlayerServiceROSBridgeComp/loadPattern "`rospack find hrpsys`/share/
hrpsys/samples/HRP-4C/data/walk2m" 1
$ rosservice call /SequencePlayerServiceROSBridgeComp/waitInterpolation
```

Using ROS node for OpenRTM users

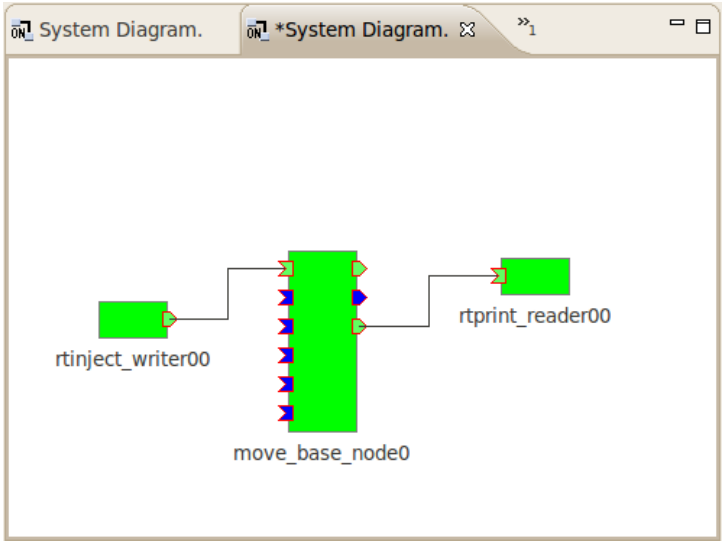
- Generate OpenRTM component from ROS node in operation

```
$ roslaunch rosnode_rtc stage_sample.launch
```

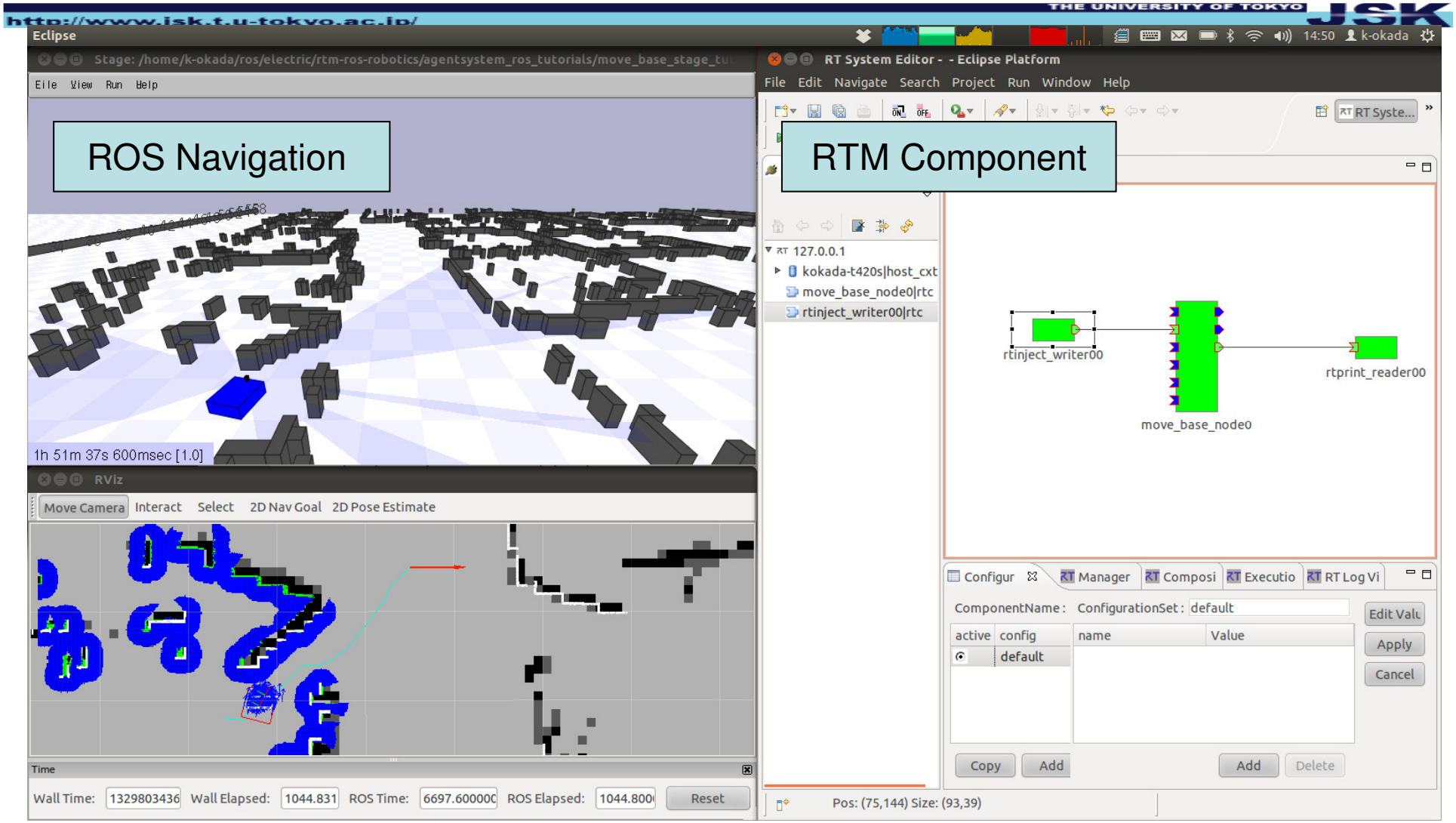
  - Analyze input/output message of the ROS node
  - Generate corresponding IDL file for OpenRTM
  - Create OpenRTM component wtth the IDL file to provide transformation between ROS message and OpenRTM data port



ROS navigation node



Auto generated OpenRTM component, that provides interfce to ROS navigation node



Launch ROS navigation tutorial

```
$ roslaunch move_base_stage_tutorial robot.launch
$ rosrun rviz rviz -d `rospack find movebase_stage_tutorial`/config/rviz.vcg
```

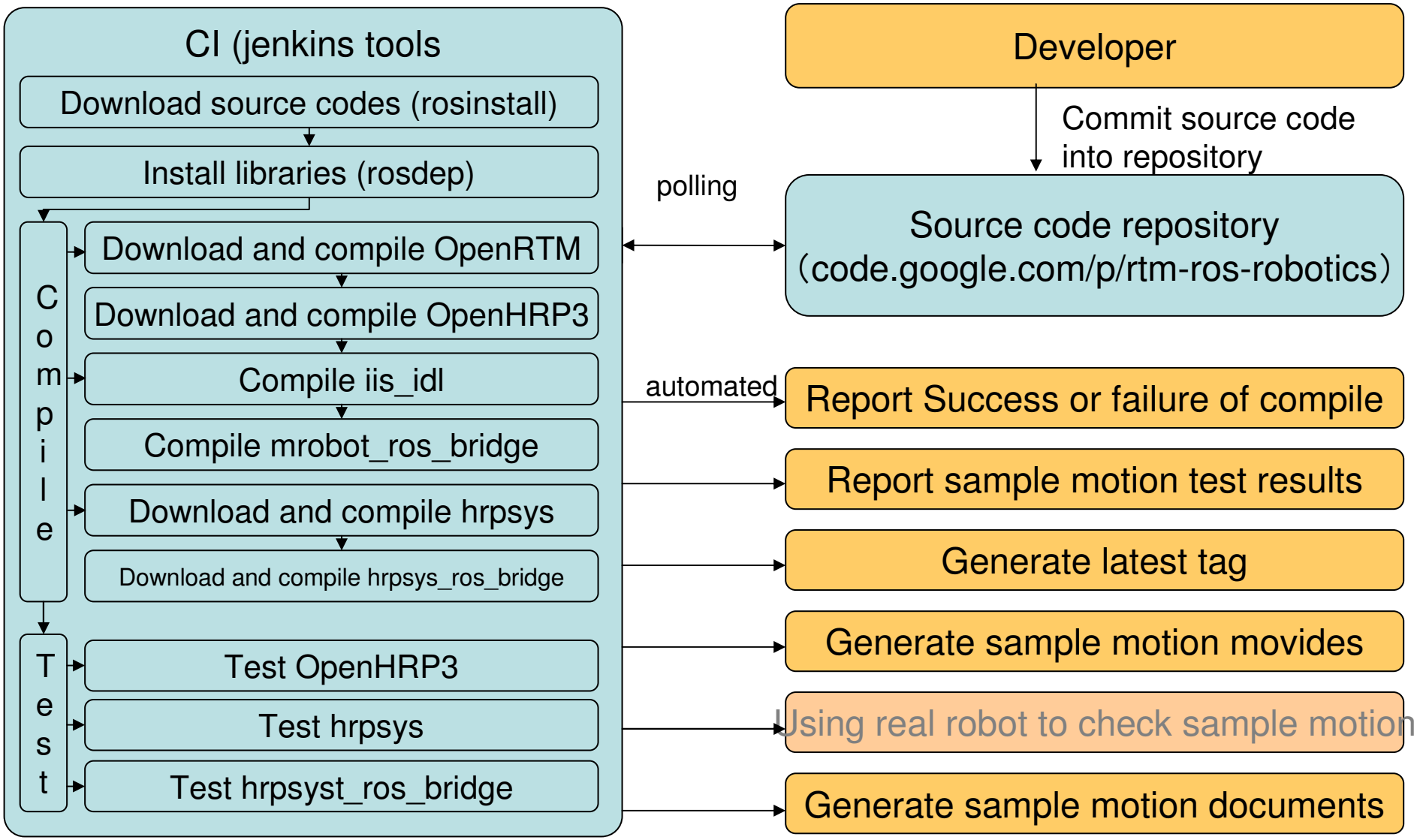
Generate OpenRTM component from ROS node

```
$ roslaunch rosnode_rtc stage_sample.launch
```

Example code to control ROS navigation through RTM data port

```
$ rosrun rosnode_rtc stage_sample_send_goal.sh
```

Documentation



Auto roslaunch documentation

- Add extra <sphinxdoc> tags intro launch file that automatically generate documentation.

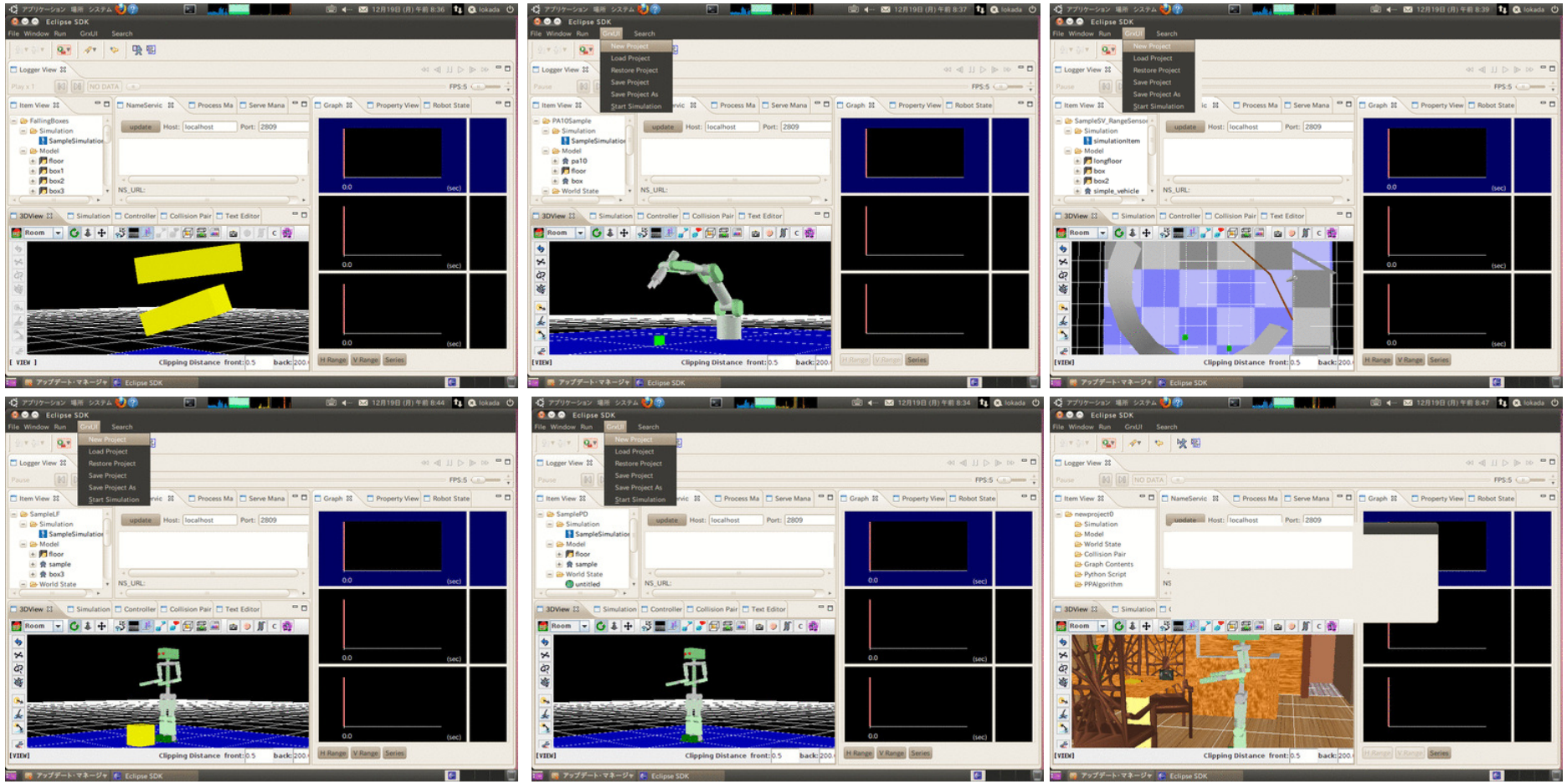
```
<launch>
  <include file=" … " />
  <include file=" … " />
   <sphinxdoc><![CDATA[
.. code-block:: bash
  rosrun roseus roseus `rospack find
  hrpsys_ros_bridge`/scripts/hrp4c-pickup.l

This launch file shows an example of ros bridge for open
  hrp4 robot…

.. video:: build/images/HRP4C_PickUp
  :width: 600
  ]]></sphinxdoc>
  <test type="test-grxui.py" pkg="openhrp3" … />
</launch>
```

This part is converted into HTML

http://rtm-ros-robotics.googlecode.com/svn/trunk/rtmros_common/hrpsys_ros_bridge/launch/hrp4c.launch
http://www.ros.org/doc/api/jsk_tools/html/

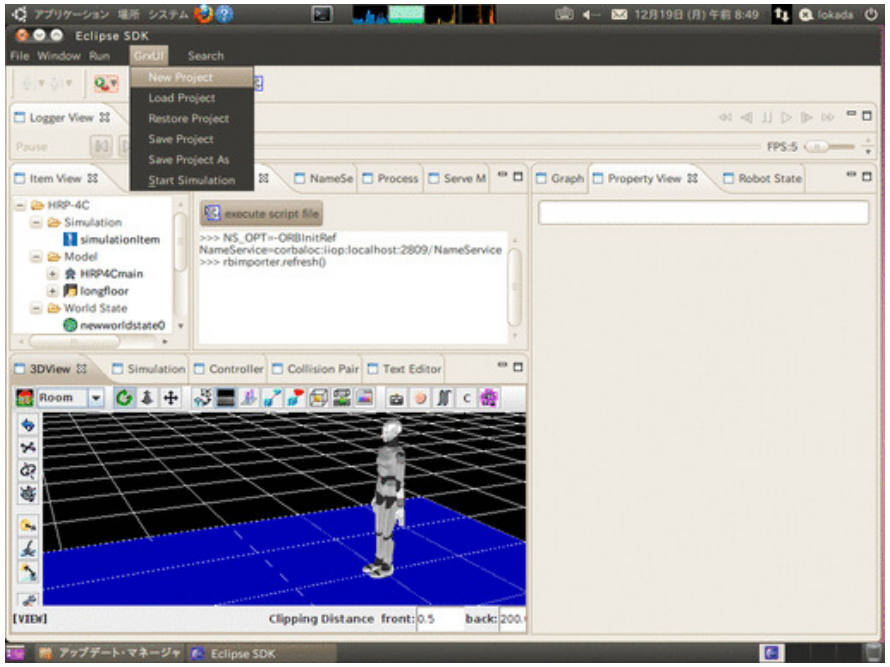
OpenHRP3 examples

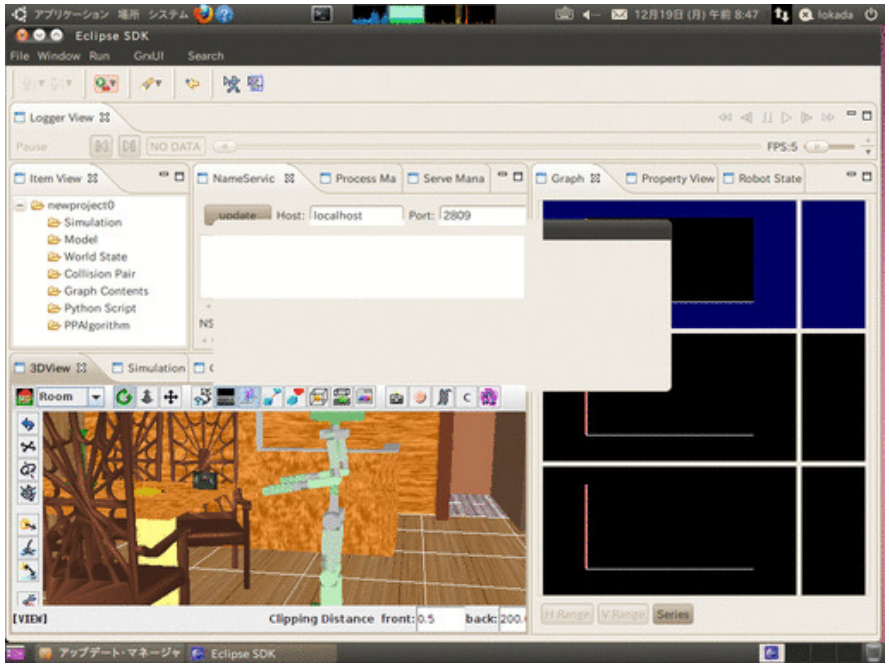
- Auto-generated mp4/ogv files from test code



hrpsys examples

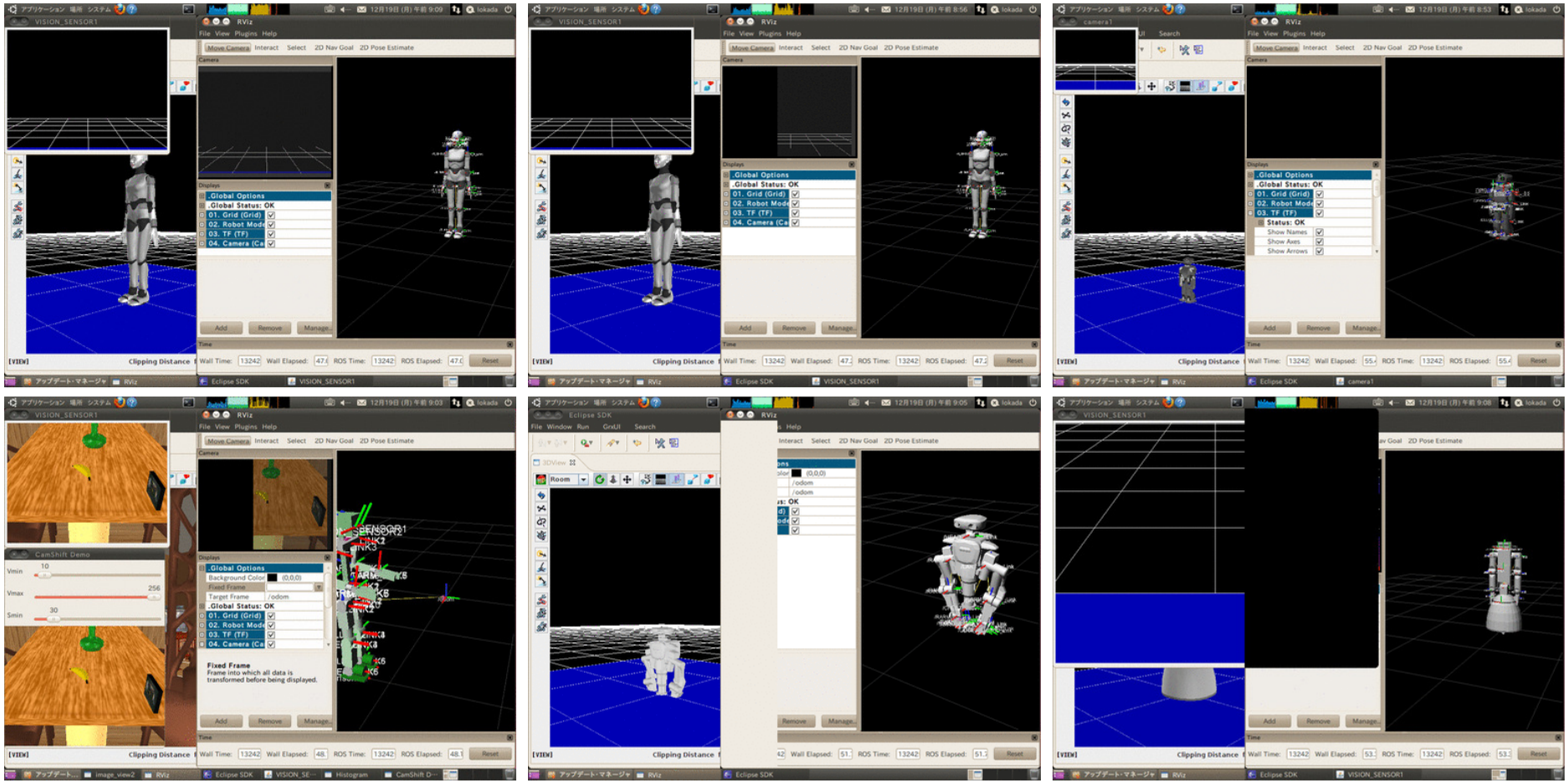
- Auto-generated mp4/ogv files from test code

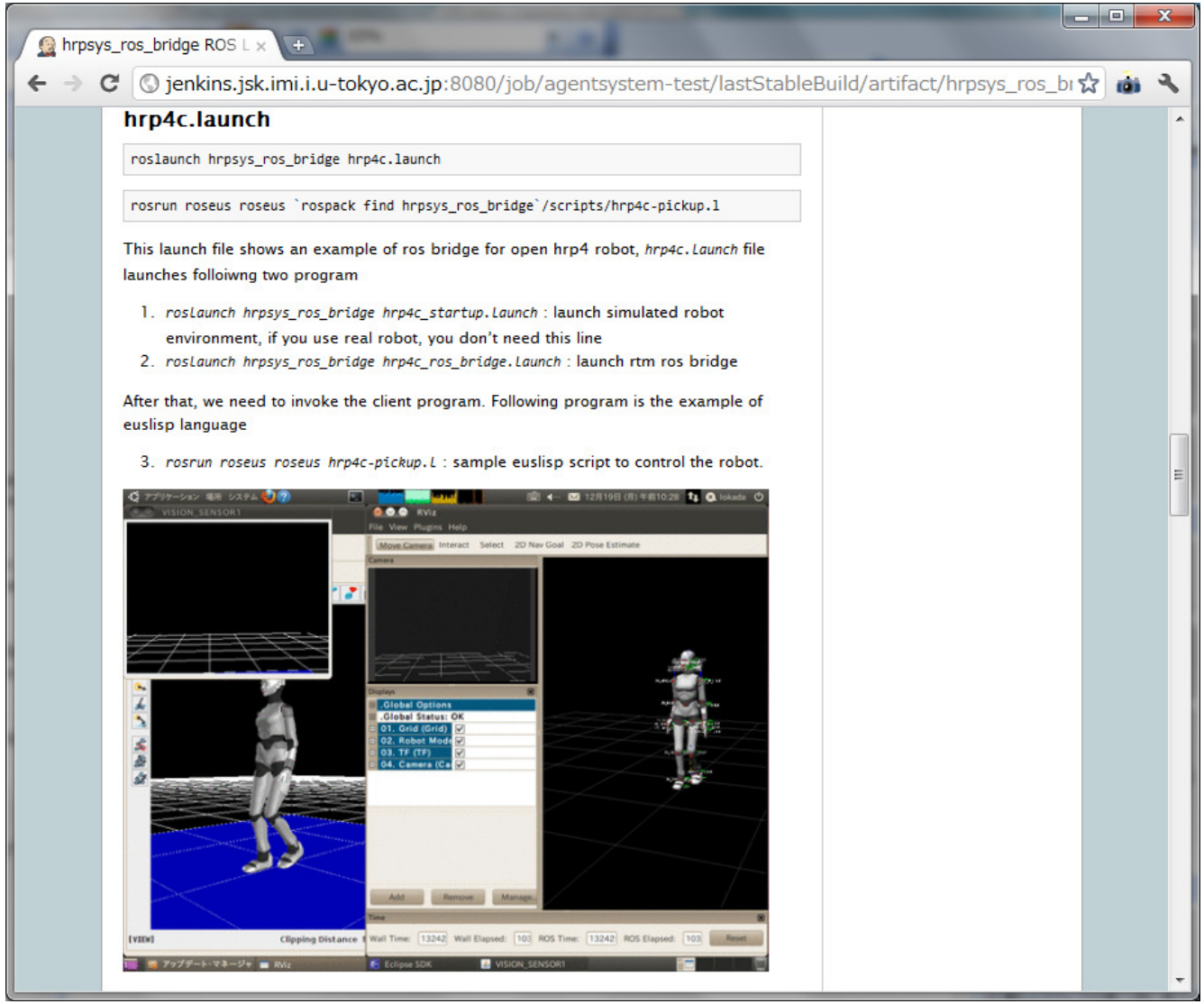


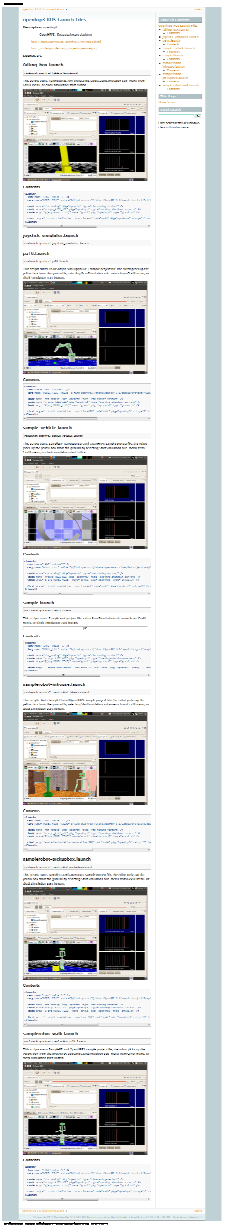


hrpsys_ros_bridge examples

- Auto-generated mp4/ogv files from test code







Example of launcdoc

http://jenkins.jsk.imi.i.u-tokyo.ac.jp:8080/job/agentsystem-test/lastStableBuild/artifact/openhrp3-example/index.html
http://jenkins.jsk.imi.i.u-tokyo.ac.jp:8080/job/agentsystem-test/lastStableBuild/artifact/hrpsys-example/index.html
http://jenkins.jsk.imi.i.u-tokyo.ac.jp:8080/job/agentsystem-test/lastStableBuild/artifact/hrpsys_ros_bridge-example/index.html

Conclusion

- Many robots, many software frameworks
- Building interoperability between OpenRTM and ROS
  - Using ROS tools to build and deploy
  - COLLADA support for robot model compatibility
  - Generating ROS node from OpenRTM component
    - Statically analyzes component (IDL) to generate service message and ros node source file
  - Generating OpenRTM component from ROS node
    - Dynamically analyzes ros node input/output message and generate OpenRTM component

svn: http://rtm-ros-robotics.googlecode.com/svn/trunk/

Issue: http://code.google.com/p/rtm-ros-robotics/issues/list

ML: http://groups.google.com/group/rtm-ros-robotics