An Integrated Distributed Simulation Environment weaving by Hakoniwa and mROS 2

Hideki Takase (The Univ. of Tokyo)
Shintaro Hosoai (The Univ. of Tokyo)
Tatsuya Fukuta (INTEC Inc.)
Mitsutaka Takada (Nagoya Univ.)
Shin Kuboaki (Change Vision, Inc.)
Takashi Mori (ESM, Inc. & Hakoniwa Lab.)
We are TOPPERS/Hakoniwa WG!!

- Hideki Takase  
  - @takasehideki  
  - The Univ. of Tokyo  
  - EMB/IoT PF, ROS

- Tatsuya Fukuta  
  - @fudekunjp  
  - INTEC Inc.  
  - Cloud, Robotics

- Shin Kuboaki  
  - @kuboaki  
  - Change Vision, Inc.  
  - MBD/MDD, App

- Shintaro Hosoai  
  - @s-hosoai  
  - The Univ. of Tokyo  
  - IDE, Modeling

- Mitsutaka Takada  
  - @mitsu48  
  - Nagoya Univ.  
  - Kernel, Automotive

- Takashi Mori  
  - @kanetugu2020  
  - ESM & Hakoniwa Lab  
  - Tech Leader!!

- About TOPPERS Project (Toyohashi OPeN Platform for Embedded Real-time Systems)
  - NPO in Japan to promote embedded systems technology and its industry by developing and releasing high-quality open-source software, especially in real-time kernels and components
  - Hakoniwa WG: working group to establish simulation technology for the IoT fields
Agenda

1. What is Hakoniwa?
2. Our Past Achievements, especially for ROS Robot
3. Latest Update: hakoniwa-mros2sim
4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
5. Wrapping Up
Agenda

1. What is Hakoniwa?
   - What’s the issue
   - Concept: HUB to Everything for Simulation!
   - Key Functions: Hakoniwa Core, Conductor and Protocol Data Unit

2. Our Past Achievements, especially for ROS Robot

3. Latest Update: hakoniwa-mros2sim

4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2

5. Wrapping Up
Issue and Background

- IoT system spans a variety of technical domains, so its development requires the collaboration of experts from different fields.
- Integration test and validation are so challenging due to these technologies’ diversity.
- When a problem unfortunately happens, investigating the cause and its path becomes intricate.
- Costs for conducting empirical experiments also increase significantly.
Let’s bring them into a BOX!!

- Arranging various technologies according to everyone's preferences
- Trying out various combinations as often we want
- Observing them according to everyone's preferences into the same box
- Bringing together things (e.g., software) from various tech fields to perform end-to-end integration, and verification on the desk!

-> Developing IoT/robot systems in a virtual environment Hakoniwa

BTY, why “Hakoniwa”?
- Hako(箱) = box
- Niwa(庭) = garden

a miniature garden, one of the traditional JP cultures (like “Bonsai”)
**Concept**

- HUB to Everything for Simulation!
  - not just a simulator, but also a framework to construct simulators
- Hakoniwa Assets: the components of the simulation target

---

Hakoniwa Core (C/C++)

Hakoniwa Daemon (for Remote Access)

Hakoniwa Library (for Physics Engines)

Hakoniwa Commands (for Test Automation)

Hakoniwa APIs (for prog. Languages)

**•** Hakoniwa Assets: components of the simulation target
Hakoniwa Core

• Functions for the seamless integration of various assets

Visualization Tool

Microcontroller Simulator

Hakoniwa Asset

Asset Management

Synchronization & Communication

Scheduling

Time Management

Hakoniwa Core

Physics Simulator

Automated Testing Tool
Hakoniwa Conductor

• Mediate simulation between assets on Hakoniwa
  • gRPC based communication between the server and clients

**Server-side**
- registration of Hakoniwa assets
- start/stop/reset of each simulation

**Clients-side**
- connect to the server as the Hakoniwa assets
Protocol Data Unit (PDU)

• Common data format to exchange in Hakoniwa
• Function design is separated to asset-dependent and -independent
• An example between Unity and ROS (topics)
  • Note: Unity and ROS are positioned as one of the "assets" in Hakoniwa
Distributed Simulation with Hakoniwa

- **Computer1**
  - Hakoniwa Proxy for PDU
  - Hakoniwa Asset

- **Computer2**
  - Server-side Hakoniwa Conductor
  - Hakoniwa Asset

- **Computer3**
  - Client-side Hakoniwa Conductor
  - Hakoniwa Asset

- **Computer4**
  - Hakoniwa Asset

Diagram illustrates the network topology and roles of the Hakoniwa components in distributed simulation.
Agenda

1. What is Hakoniwa?
2. Our Past Achievements, especially for ROS Robot
   - 4 Prototypes to Ensure the Hakoniwa’s Concept
   - hakoniwa-ros2sim: ROS controlled Robot Simulation
   - Actual Use Cases
3. Latest Update: hakoniwa-mros2sim
4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
5. Wrapping Up
Hakoniwa Prototypes Previously Achieved

Microcomputer-controlled robot simulation

Collaborative simulation of multiple robots

Integration with reinforcement learning

ROS controlled Robot Simulation
ROS controlled Robot Simulation

- Docker for the platform
  - pre-built container image for building & executing ros2app
  - able to use on multiple environments (Windows/WSL2 as the default)
- Unity for physics and visualization
  - TurtleBot3 as the reference
  - ROS-TCP-Endpoint for assets communication
Actual Use Cases

• Athrill (micro-controller simulator) for ET-robocon [https://www.etrobo.jp/](https://www.etrobo.jp/)
• Group exercises (PBL) in some Japanese universities
• Automotive software education for embedded engineers [https://github.com/ncesnagoya/autosar_os_training](https://github.com/ncesnagoya/autosar_os_training)
Agenda

1. What is Hakoniwa?
2. Our Past Achievements, especially for ROS Robot
3. Latest Update: hakoniwa-mros2sim
   - Introduction of mROS 2
   - [FYI] ROSCon JP 2023 Workshop
   - hakoniwa-mros2sim: Simulator for mROS 2 app by Hakoniwa Tech.
4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
5. Wrapping Up
What is mROS 2?

mROS 2 application

mROS 2 API

mROS 2 comm. lib.

RTPS (embeddedRTPS)

UDP stack (lwIP)

CMSIS wrapper

Kernel (ASP3, Mbed, POSIX)

HAL library

Embedded devices

pub/sub messaging for Topic partially compliant with rclcpp

autonomous communication in accordance with RTPS specification

lightweight and efficient process by C/C++ for <~200MHz / <~1MB

real-time kernels for EMB contributes perf. and mem. usage

mROS 2 on EMB board
better perf. & mem. usage partially compatible with rclcpp
only for Topic comm., and many unsupported features such as QoS, Service,

https://vimeo.com/showcase/9954564/video/767140724
Currently Supported

- mROS-base/mros2-as3-f767zi
  reference implementation of mROS 2 for STM32
  NUCLEO-F767ZI with TOPPERS/ASIP kernel
  ★ 16  ¥ 2

- mROS-base/mros2-mbed
  reference implementation of mROS 2 for Mbed OS
  ★ 40  ¥ 6

- mROS-base/mros2-esp32
  reference implementation of mROS 2 for ESP32 boards
  ★ 27  ¥ 3

- mROS-base/mros2-posix
  reference implementation of mROS 2 for POSIX layer
  ★ 4  ¥ 1

NEW!!

freeRTOS

POSIX
ubuntu

watch later!!
News!!

ROSCon JP '23 contents is public!!
(only in JP:D)

https://roscon.jp/2023/#workshop
hakoniwa-mros2sim :D

- Target: mros2 app & robot

Pi:Co Classic3 (MicroMouse)

mROS-base/rcjp2023_mros2
https://github.com/mROS-base/rcjp2023_mros2
Agenda

1. What is Hakoniwa?
2. Our Past Achievements, especially for ROS Robot
3. Latest Update: hakoniwa-mros2sim
4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
   - Hakoniwa’s Future Vision for Robotic Service
   - Expected System in the Future and its Architecture Design
   - Integration of mROS 2 into Hakoniwa Communication
   - Demonstration!!
5. Wrapping Up
Our Future Vision for Robotic Service

How to Integrate?
- AI Analysis Space
- Cloud Service

What to do?
- Generative AI Agent Space
- Gaming Space
- Virtual Service

How to Integrate?
- Robot Service Networks
- Micro Controller

Coverage of ROS
- Robot Controller

A new trend!!
Challenges faced in the embedded systems domain
Challenges faced in the cloud domain
Coverage of ROS

How to Integrate?

What to do?
Fusion of Virtual and Real

• Integration of Virtual and Real in Simulation

Virtual

Generative AI Agent Space

AI Analysis Space

Gaming Space

Real

Robot
Expected System in the Future

Virtual World
- AI Agent
- Virtual People (Digital Avatar)

Real World
- AR Agent
- People
- Real Robot

Virtual Signal
- Virtual Robot (Digital Twin)
Architecture Design: Overview

Real World
- mROS
- Infrastructure
- Sensor
- AR Device
- Real Robot
- Actuator

Central Server (ROS)

Wireless Network

Cloud Network (MQTT)

Virtual World
- Virtual Robot
- Virtual People
- Virtual Signal
- AI Agent
- Virtual Service
- Virtual/Real Map
- Cloud Server

Virtual/Real Map

AR Device
but Why mROS 2?

- Integration of mROS 2 communication into Hakoniwa!
  - lightweight but direct communication on RTPS
  - mros2-posix can run on general-purpose OSes (even on macOS)
  - support mros2 communication as one of Hakoniwa PDUs!!
Demonstration!!
Agenda

1. What is Hakoniwa?
2. Our Past Achievements, especially for ROS Robot
3. Latest Update: hakoniwa-mros2sim
4. Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
5. Wrapping Up
Wrapping Up!

- Introduction of Hakoniwa (箱庭)
  - HUB to Everything for Simulation!
    not just a simulator, but also a framework to construct simulators
  - hakoniwa-ros2sim: our past achievement for ROS robot
  - hakoniwa-mros2sim: new simulator for mros2 robot
- Fusion of Virtual and Real weaving by Hakoniwa and mROS 2
  - Integration of mROS 2 into Hakoniwa Communication
  - Expected System in the Future and its Architecture Design
- What’s Next??
  - Open up the future by increasing the adoption in actual development scenes
  - and, enrich documentation, especially in English ;(