Model-based design of Intelligent Mobile Robot

Takahiro TAKASU*, Tsunehiko FUJITA*, Yusuke ZAMA*,
Koji ISHIDA*, Makoto MIZUKAWA*, Yoshinobu ANDO*,
Takashi YOSHIMI*, Takeshi SAKAMOTO**

*Shibaura Institute of Technology, Japan
**Technologic Arts Incorporated, Japan

Introduction

- Intelligent RT Software Project(NEDO project)
  - The objective this project is to provide modularized intelligent robot software
  - Plan to develop personal mobility service
    - Safe and convenient transportation service for every people

Introduction

- Intelligent RT Software Project(NEDO project)
  - For efficiency robot development
  - Promoting RT-Middleware(RTM)

    •Common platform of robot system development
    •Component based system
    •Providing OMG standard interfaces of RT-Component(RTC)
    (RT-Component = Robotic software elements)

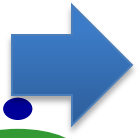
    - Developing useful RT-Components
  - Applying RTM to experimental platforms

Introduction

- Intelligent RT Software Project(NEDO project)
  - Providing reusable RTCs
    - Defining common interfaces
    - Established RTC-Center to maintain RTCs
      - Accumulating RTCs
      - Users can select useful RTCs
      - Developers can receive feedbacks from users

Problem

Activity of extending reusability of RTC



Not discussing about reusability of “Platforms”

Purpose of our research

Development of intelligent mobile robot using model-based design

Derive reusable and versatile robot model

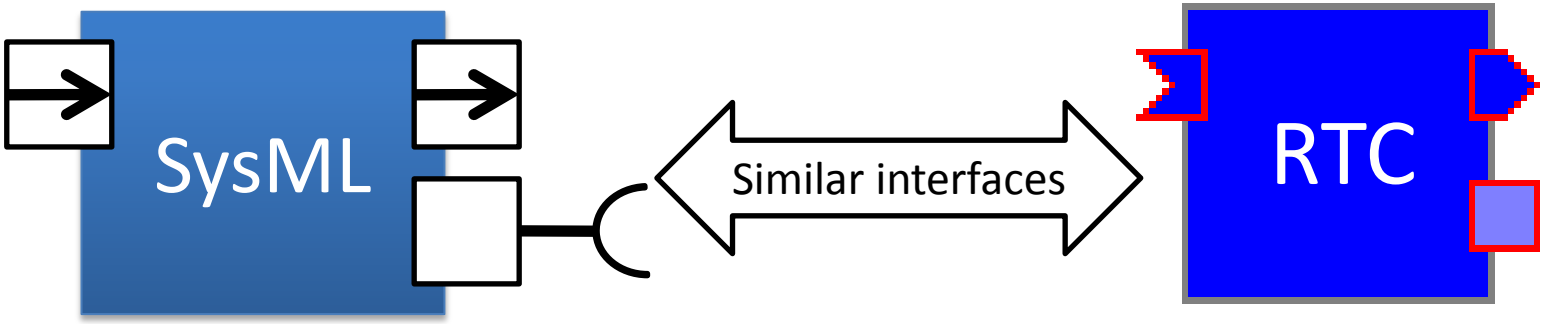
→ Making models independent from physical robot specific

Employing existing -RTCs

→ Making effortless transition from model to real system

Our development

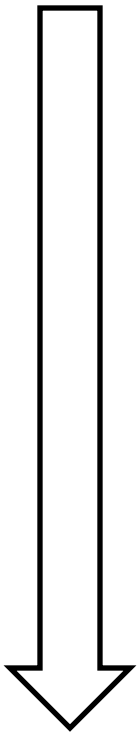
- Our implementation platform
  - OMG System Modeling Language(SysML)
  - RT-Middleware



We can replace SysML blocks to RTCs

Modeling

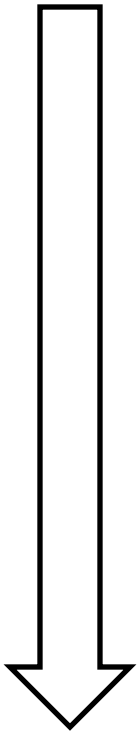
Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Modeling

Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Operating environment analysis

What are objects surrounding robot?

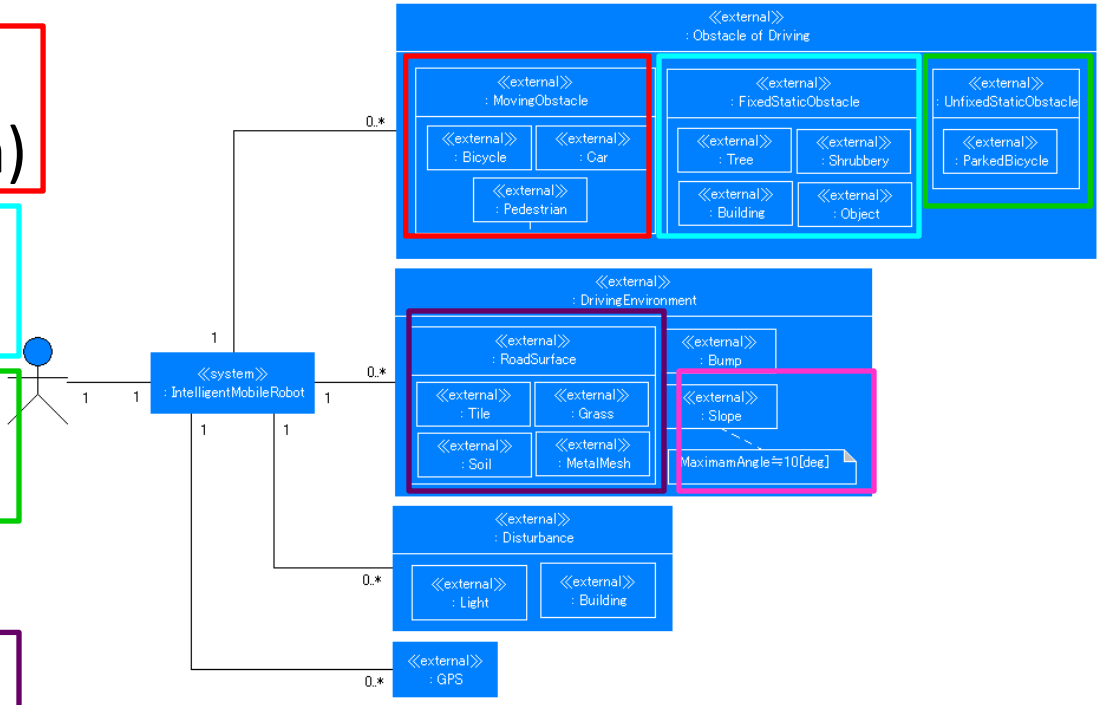
How do they affect robot operation?

Categorizing objects and environments

For deriving requirements of robot functions

Operating environment analysis

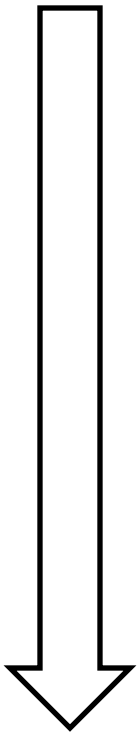
- Various obstacles
  - Moving obstacles (Bicycle, Car, Pedestrian)
  - Fixed static obstacles (Tree, Building, Object)
  - Unfixed static obstacles (Parked Bicycle)
- Driving environment
  - Different road surfaces (Tile, Grass, Soil)
  - Slope (Max 10[deg])



Context Diagram

Modeling

Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Functional requirement analysis

What are problems that mobile robots have to deal with?

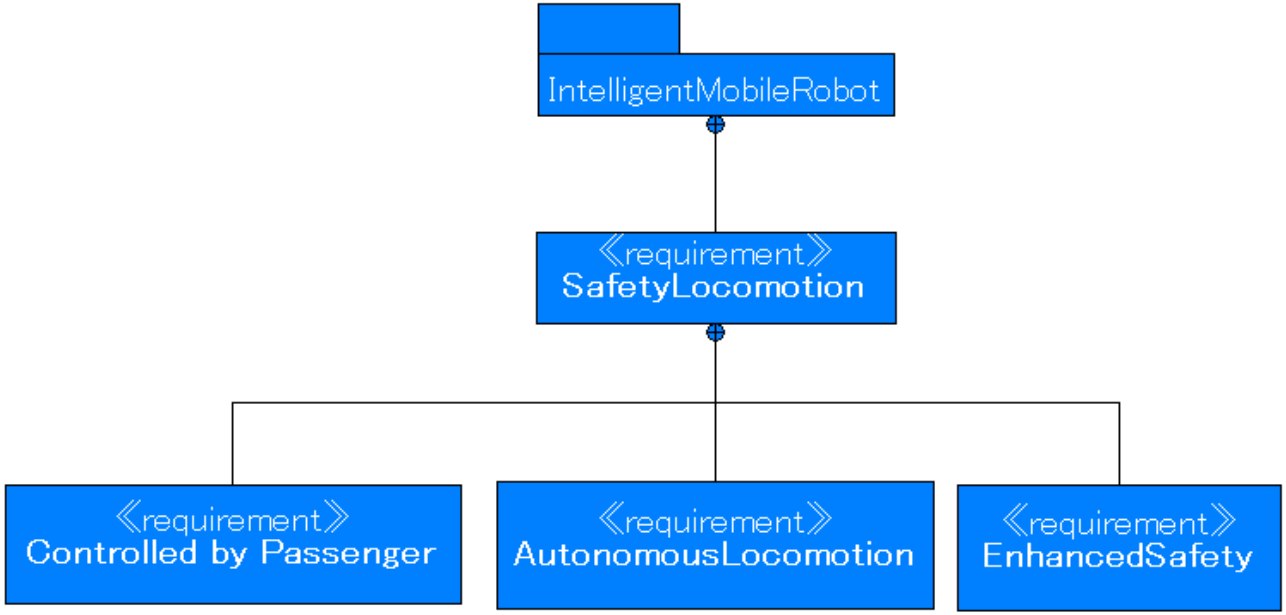
- Deriving from “Operating environment analysis”
  - Various obstacles
  - Driving environment
    - Different road surfaces
    - Slope (Max 10[deg])

This step

Clarify problems of operation

Functional requirement analysis

Derive from “Operating environment analysis”

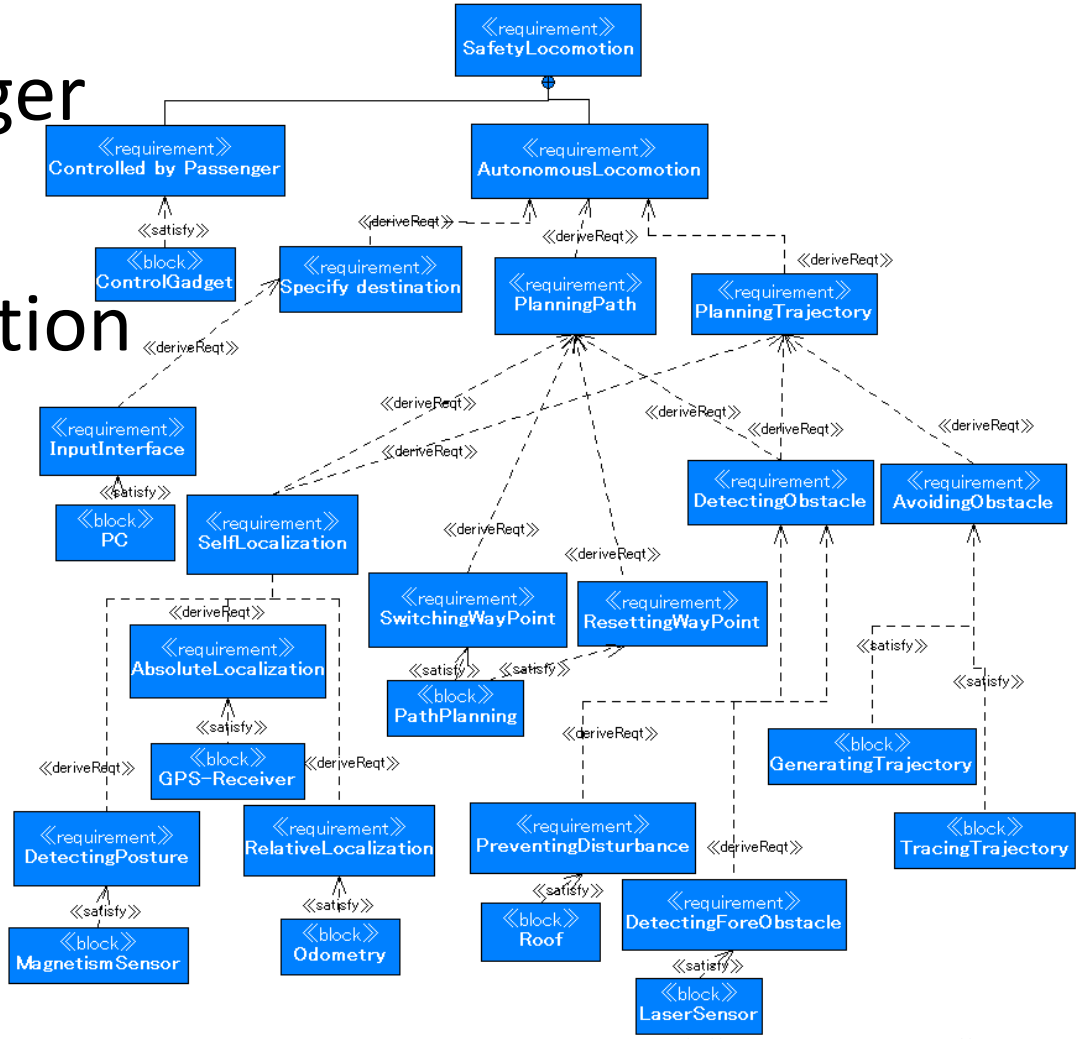


Basic functional requirements

Three important basic requirements for mobile robot

Functional requirement analysis

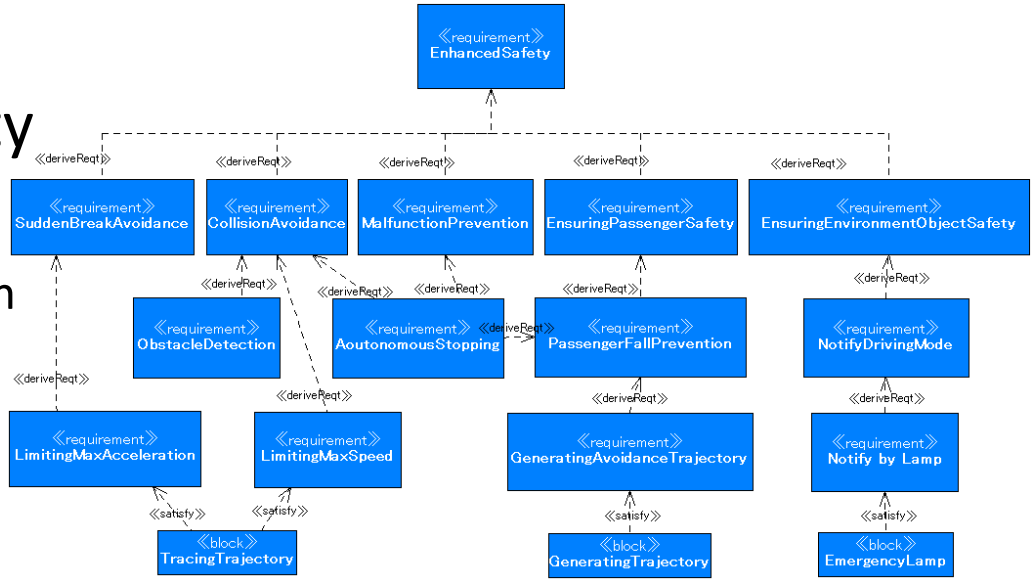
- Controlled by Passenger
  - Control Gadget
- Autonomous Locomotion
  - Specify destination
  - Path Planning
  - Trajectory Tracing
  - Self Localization
  - Obstacle Avoidance



Functional requirements of “Locomotion”

Functional requirement analysis

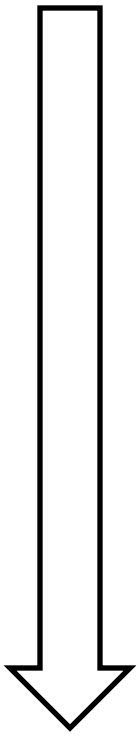
- Enhanced Safety (abridged)
  - Ensuring Environment Object Safety
    - Notify Driving Mode
      - Emergency Lamp
  - Ensuring Passenger Safety
    - Passenger Fall Prevention
      - Limiting Max Acceleration
  - Malfunction Prevention
      - Intelligent Battery
      - Safety Monitoring
      - Emergency Switch



Part of functional requirements of “Enhanced Safety”

Modeling

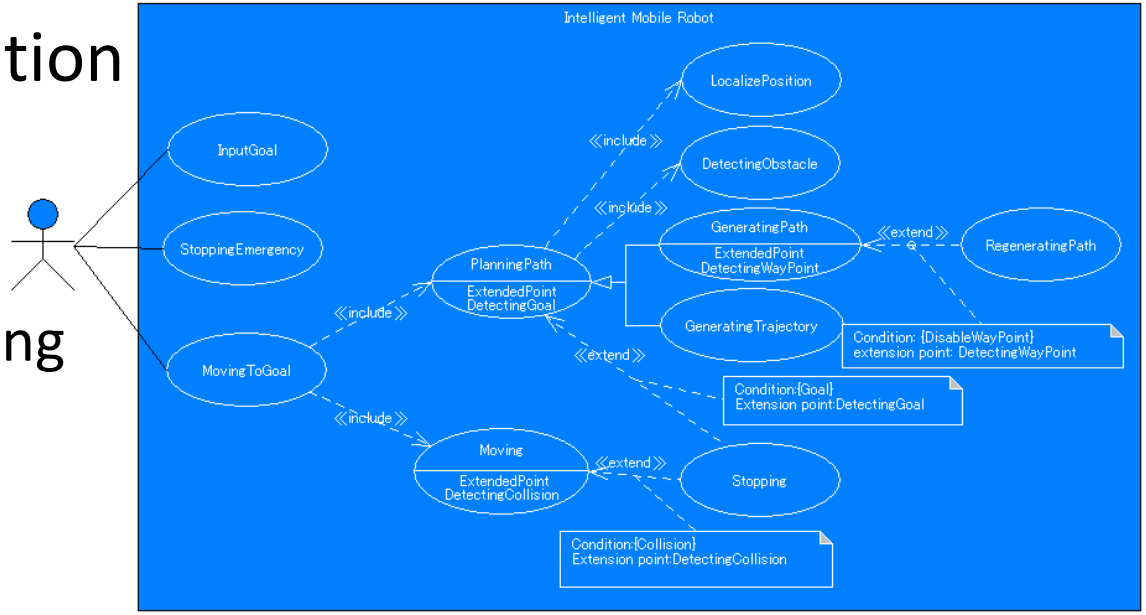
Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Necessary functions identification

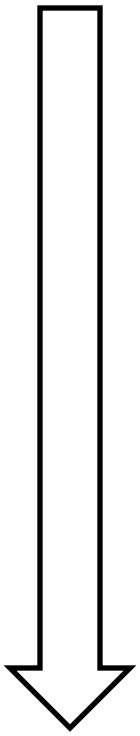
- Core functions of intelligent mobile robot
  - Specify destination
  - Emergency Stop
  - Navigate to destination
    - Path Planning
    - Path Generating
    - Trajectory Generating
    - Obstacle Detecting
    - Position Localizing
    - Errors Detecting



Use case diagram of our robot

Modeling

Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Hardware configuration

- Necessary function implementation devices
  - Vehicle
    - Driving outdoor environment

Chasswheel Four X

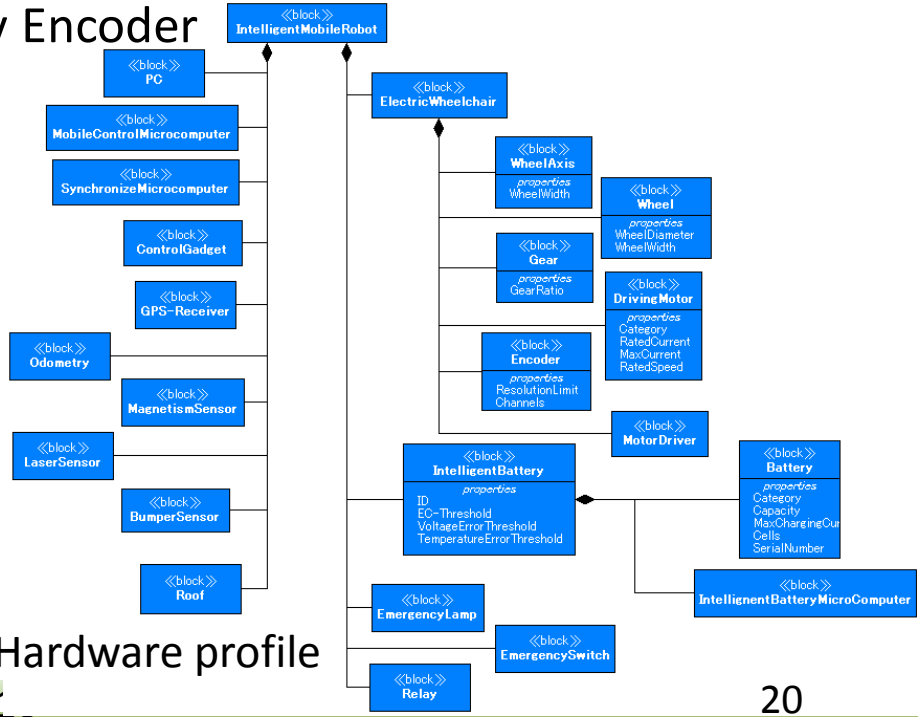
Using as chassis

Our robot

We adopted “Four X” for driving outdoor environment

Hardware profiling

- Necessary function implementation hardware configurations
  - Various sensors
    - For localization self position
      - GPS, Magnetism, Gyro, Rotary Encoder
    - For obstacle avoidance
      - Laser sensor, Bumper sensor
    - For battery malfunction
      - Intelligent battery
  - Emergency Rump
  - Emergency Switch



Hardware profile

Software structuring

What software components do we need?

Deriving from “Necessary function analysis”

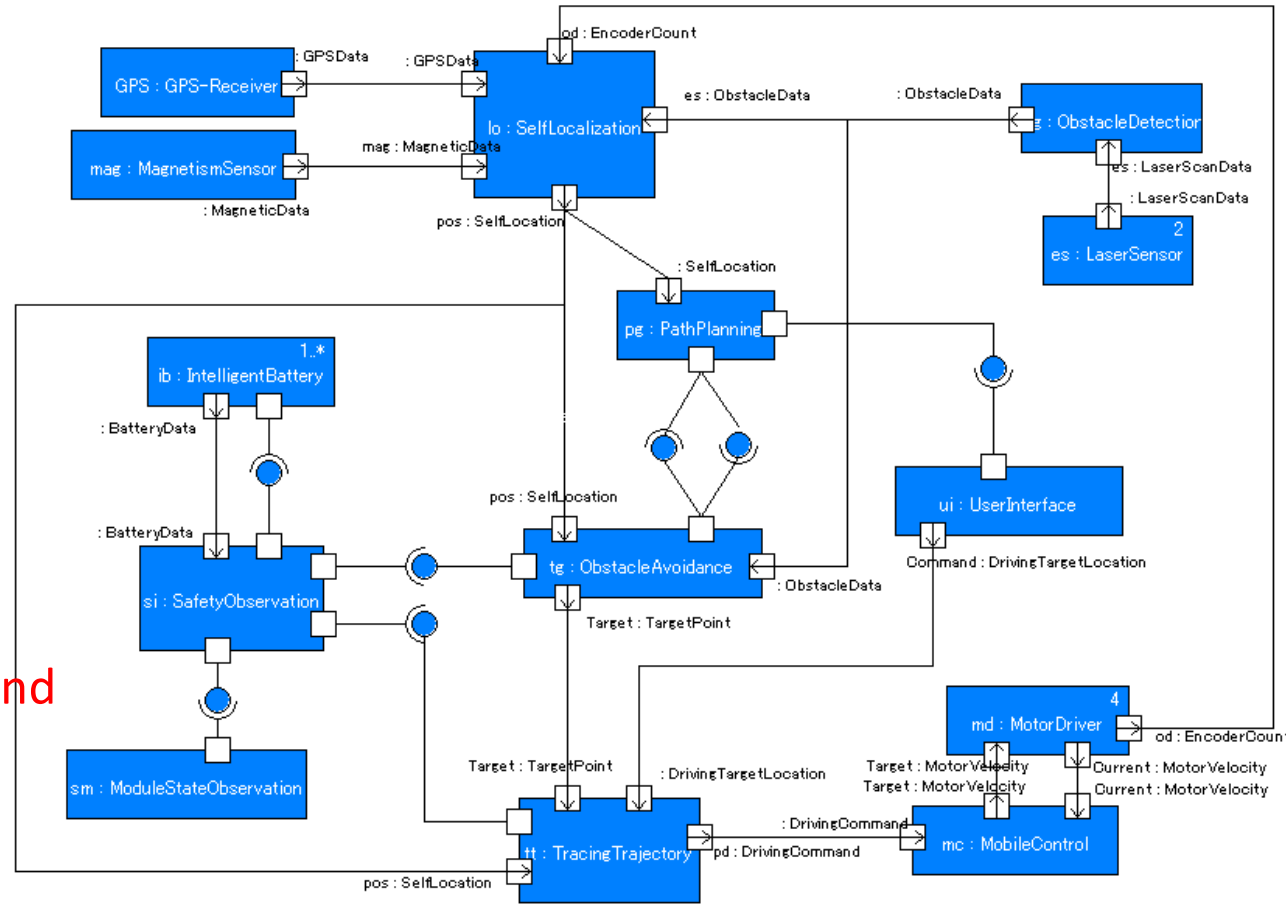
How to connect and communicate between software components?

What are functionalities of each components?

Organizing them from “Necessary function analysis”

Data profiling

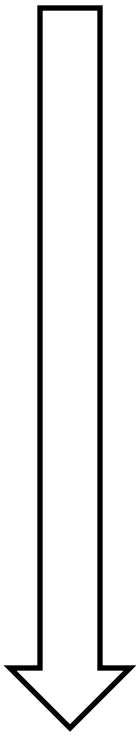
- Self localization
  - Self position
- Obstacle detection
  - Obstacle data
- Path planning
  - Next path
- Obstacle avoidance
  - Trajectory
- Tracing trajectory
  - Target velocity
- Mobile control
  - Motor driver command
- Safety observation
  - Finding error
- User interface
  - Robot command



Software configuration

Modeling

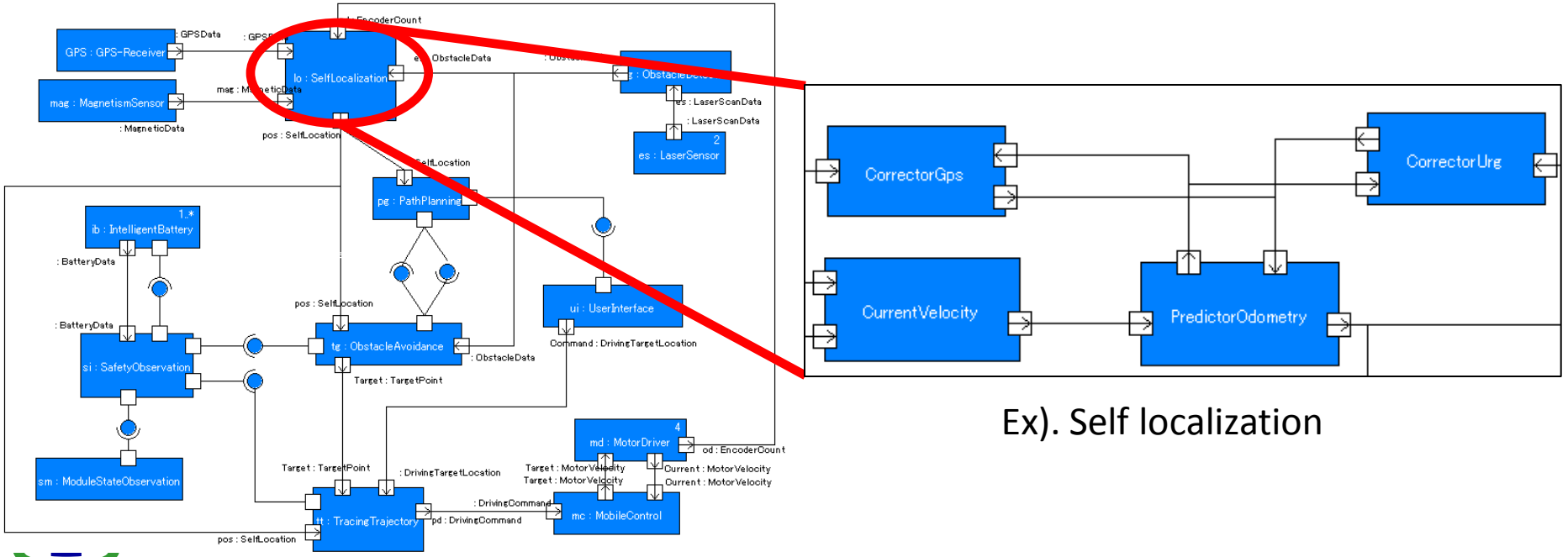
Our development is promoted by following steps



- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

Model to RTCs Mapping

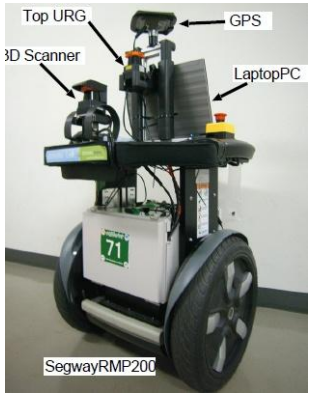
- Replacing software blocks with RTCs
  - Each software elements is composed of some RTCs
  - Selecting suitable RTC for functionality of each component



Ex). Self localization

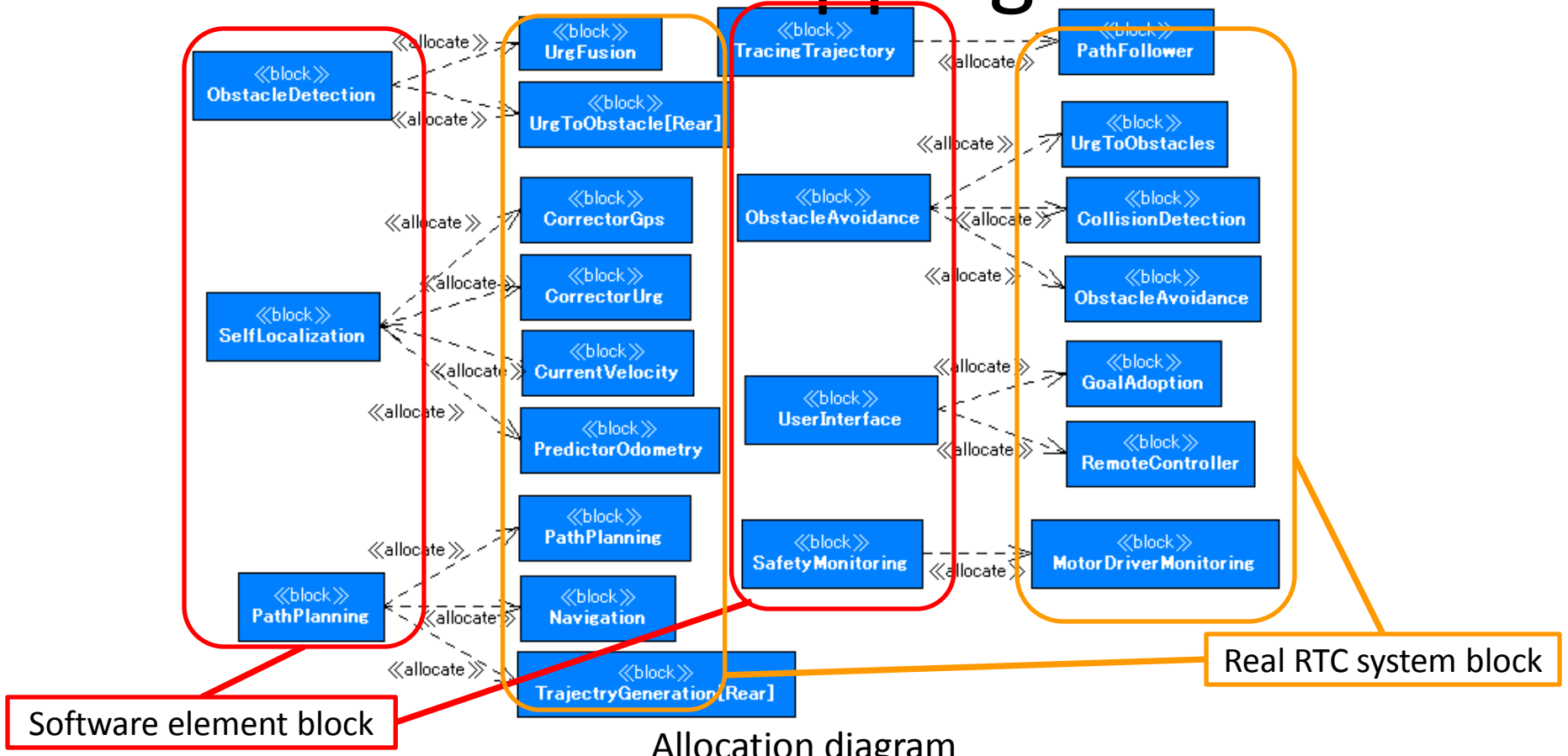
Model to RTCs Mapping

- Tohoku Univ. RT-Components
  - ✓Completed the whole course in
    “Real World Robot Challenge 2009”
    - ☐Robust navigation of outdoor environment
  - ✓Similar algorithm
    - ☐Using GPS map
    - ☐Self localization by GPS and Odometry
    - ☐Obstacle avoidance by Laser Range Finder



Segway RMP 200

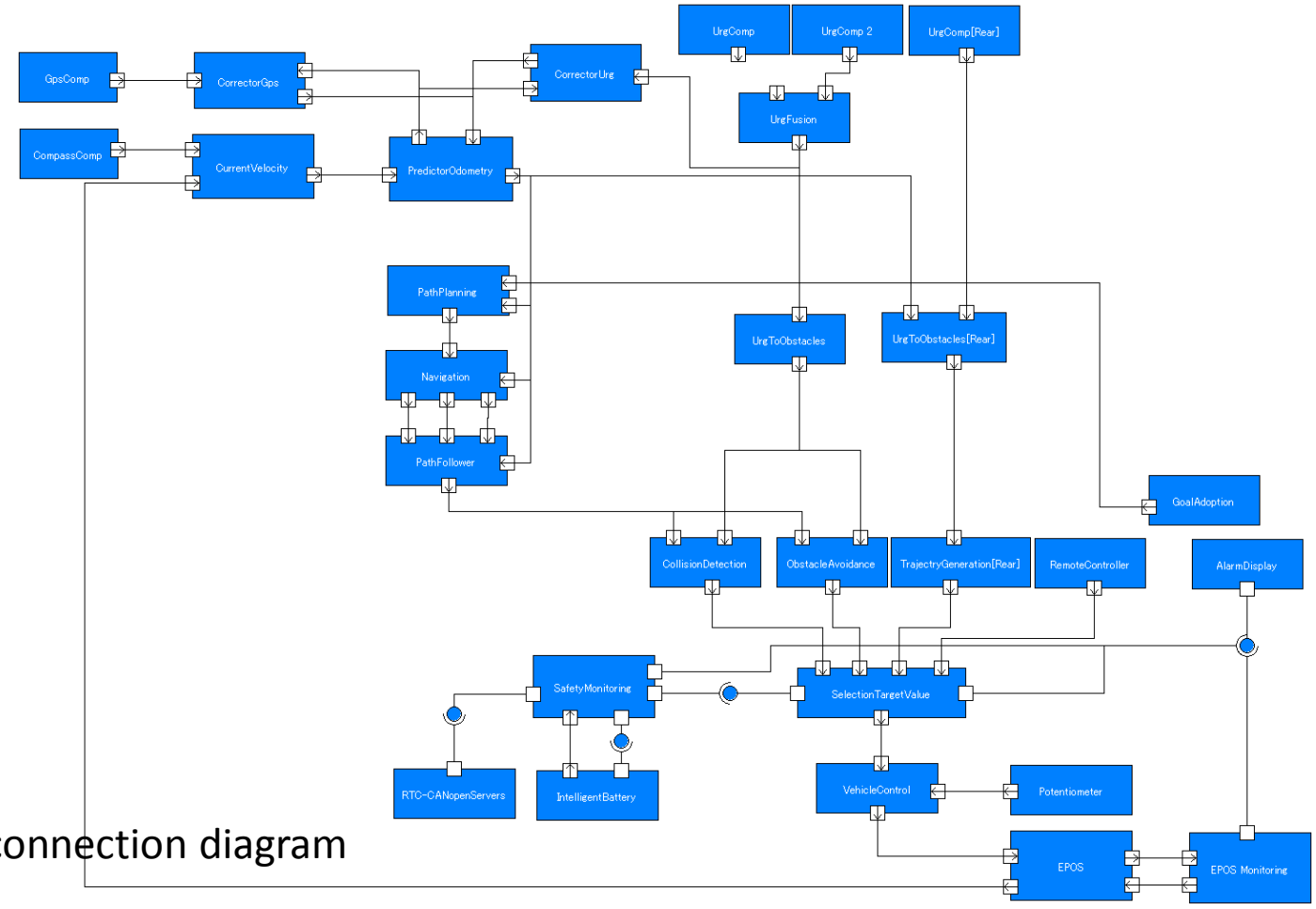
Model to RTCs Mapping



These blocks have clear functionalities

Replaced by other RTCs with different specification or parameters

Model to RTCs Mapping

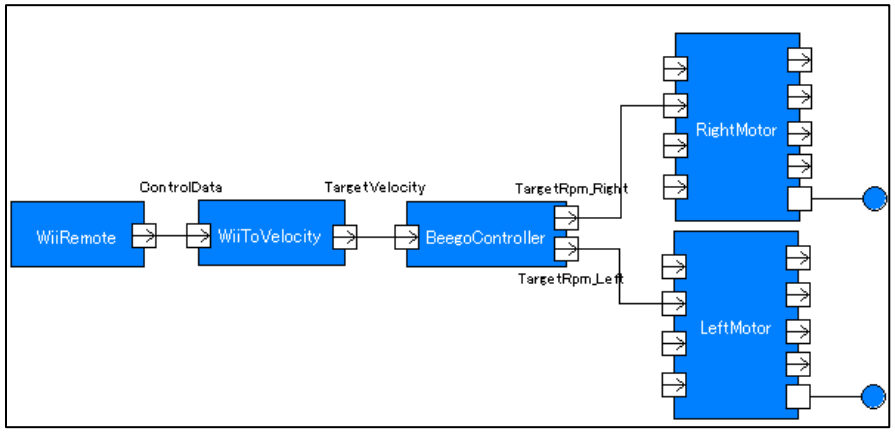


RTC connection diagram

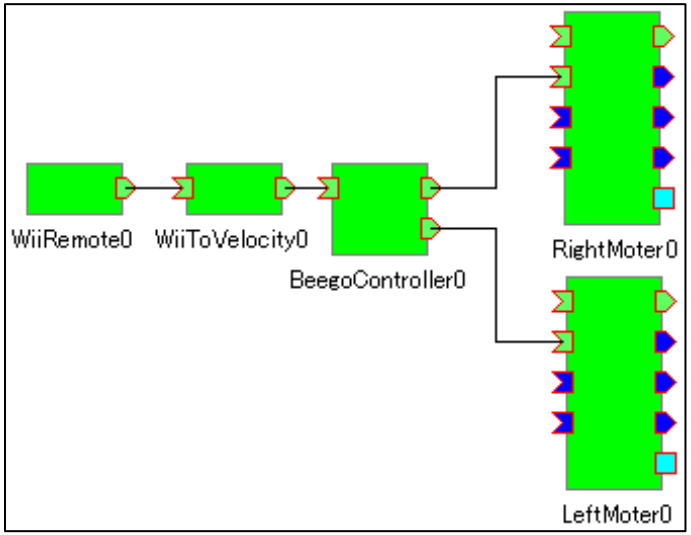
This mapping reduces development effort

Model to RTCs Mapping

Ex). Model to RTCs Mapping of small robot system



SysML model



RTC system

We will replace our model to real RTCs system

Future works

This presentation

- Operating environment analysis
- Functional requirement analysis
- Necessary functions identification
- Hardware and Software configuration
- Model to RTCs Mapping

This presentation is only basic analysis of development

We would plan to apply this model to real robots

Conclusion

- We proposed model-based development of Intelligent Mobile Robot.
- This presentation showed progressive development of robot.
- SysML model-based development helps to structurize RTC-system.

Model-based design of Intelligent Mobile Robot

Takahiro TAKASU*, Tsunehiko FUJITA*, Yusuke ZAMA*,
Koji ISHIDA*, Makoto MIZUKAWA*, Yoshinobu ANDO*,
Takashi YOSHIMI*, Takeshi SAKAMOTO**

*Shibaura Institute of Technology, Japan
**Technologic Arts Incorporated, Japan