# 지능형 로봇 소프트웨어 무결점 검증기 개발

박종우
서울대학교 로봇자동화연구실

# Present Status of Intelligent Robot Softwares

Seoul National University
School of Mechanical and Aerospace Engineering

ROBOTICS LABORATORY

# Difficulty in Researching

- Robotics Research
  - From a tiny bolt to a graphical user interface.
  - Hardware requires huge resources of money.
  - Software requires vast man powers.

- Not-reusable Resources
  - General standards of hardware and software does not exist yet.
  - Hardware and software are generally not reusable, especially software is not.
  - Even, small changes of a hardware cause big changes of a corresponding software.

# Robot Software Development

- Implement basic math library to high-level controller.
- Develop each of functional modules—sensing, actuating, decision, etc. and integrate them.

- Simulation environment is essential.
  - Hardware is expensive.
  - Unverified controller is dangerous to an operator as well as a robot.
  - However, building a simulation environment is another burden.

- Integrated environment are required.

# Efforts for Integrated Environment

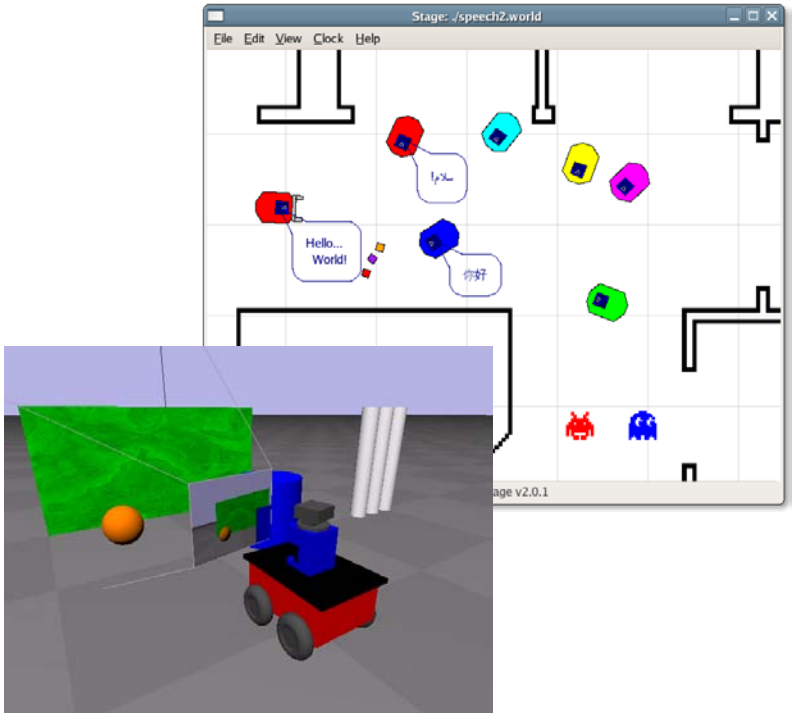
- From early 2000's.
  - Player/Stage/Gazebo, USA
  - OpenHRP, Japan
  - Orocos, EU
  - RUPI, 차세대 로봇 소프트웨어 플랫폼 사업, 한국

  and so forth.

- Still there's no leading group.

# Player/Stage/Gazebo

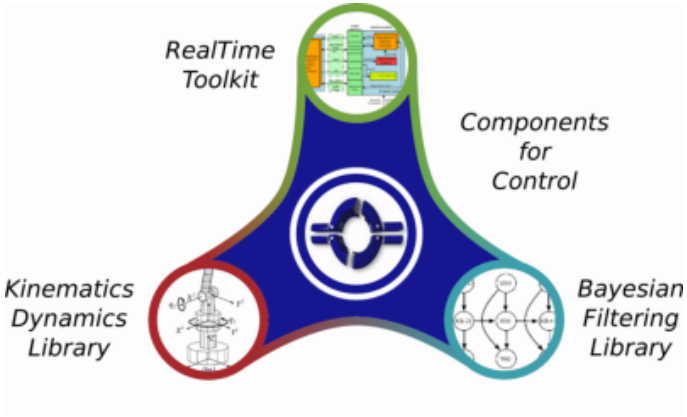
- Open source, cross platform.
- Player
  - Network server for robot control.
  - Provide an interface to the robot's sensors and actuators over the IP network.
- Stage
  - 2D kinematics simulation environment.
- Gazebo
  - 3D dynamics simulation environment.



# OROCOS
## Open Robot Control Software

- A general-purpose, free software, and modular framework for robot and machine control.

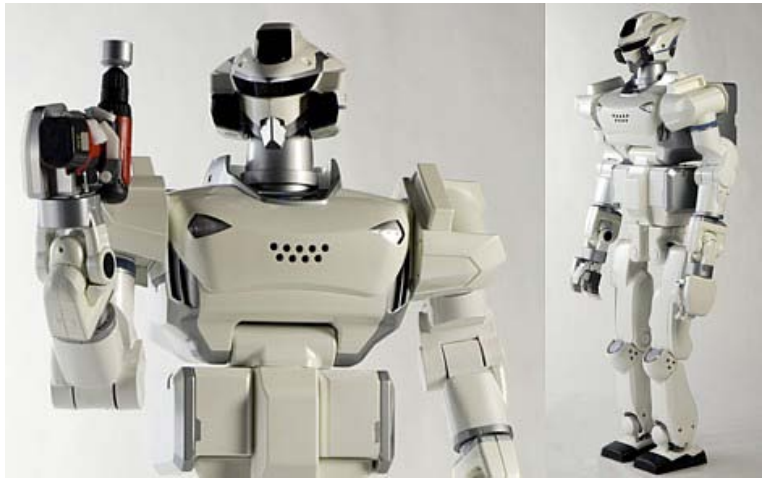
- 4 C++ libraries:
  - Real-time toolkit.
  - Kinematics and dynamics library.
  - Bayesian filtering library.
  - Orocos component library.



# OpenHRP
## Open Architecture Humanoid Robotics Platform

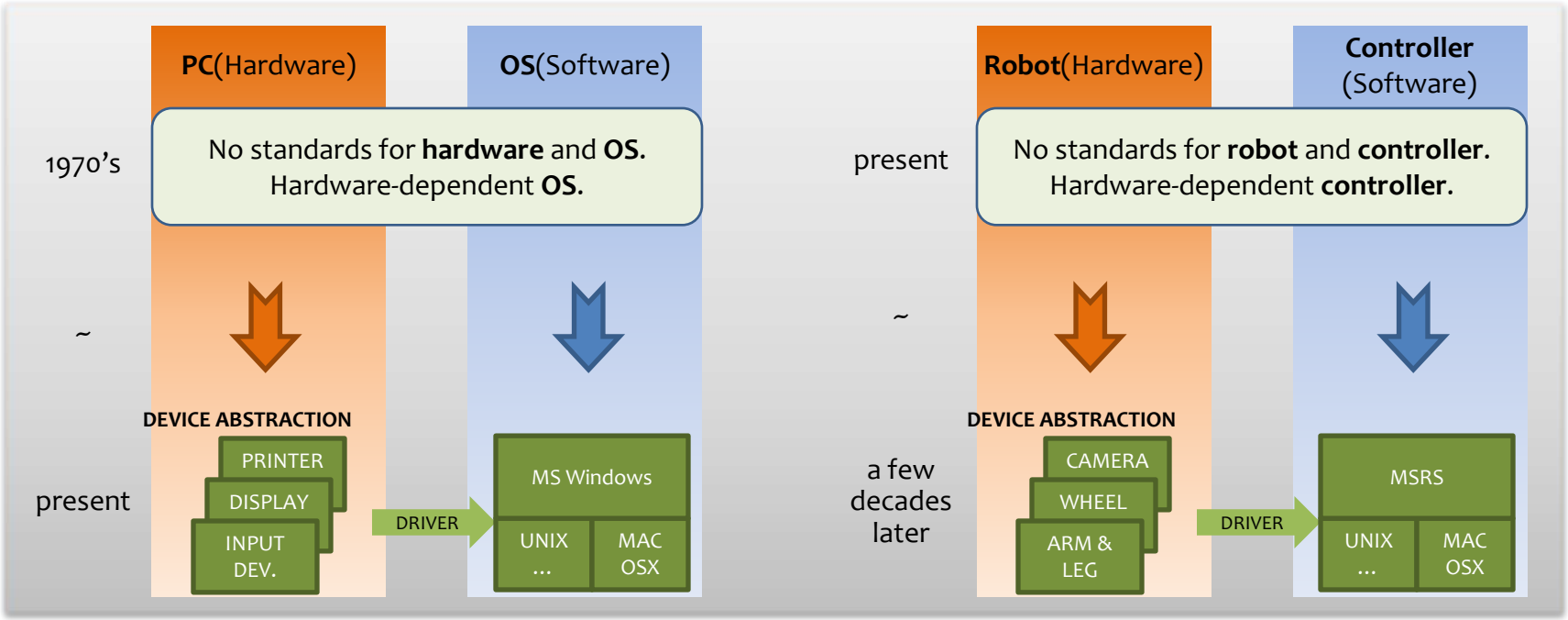
- Software platform for humanoid robotics.
  - Dynamics simulator.
  - View(camera) simulator.
  - Motion controllers.
  - Motion planners of humanoid robots.

- Integrated with CORBA.



# Emergence of MSRS

- Dec 2006, Microsoft released Microsoft Robotics Studio a.k.a. MSRS.
- Various robot software platforms have emerged.



- Robot control softwares are about to explode.

- However, Player/Stage/Gazebo, Orocos, OpenHRP, etc. are not proper for software verification, since they are environments, not robot controllers.

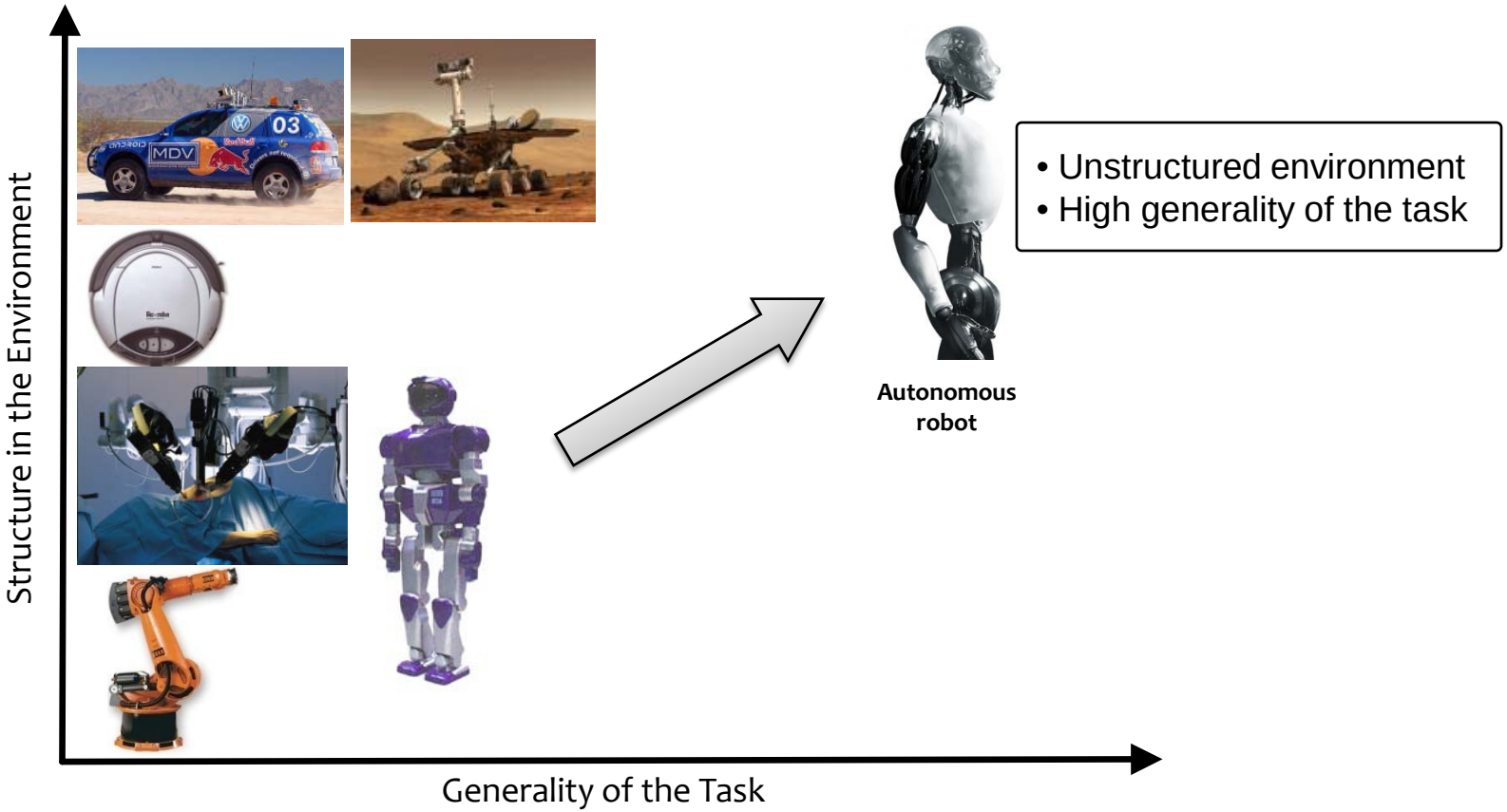
# Motion Planner for Mobile Manipulator

Seoul National University
School of Mechanical and Aerospace Engineering

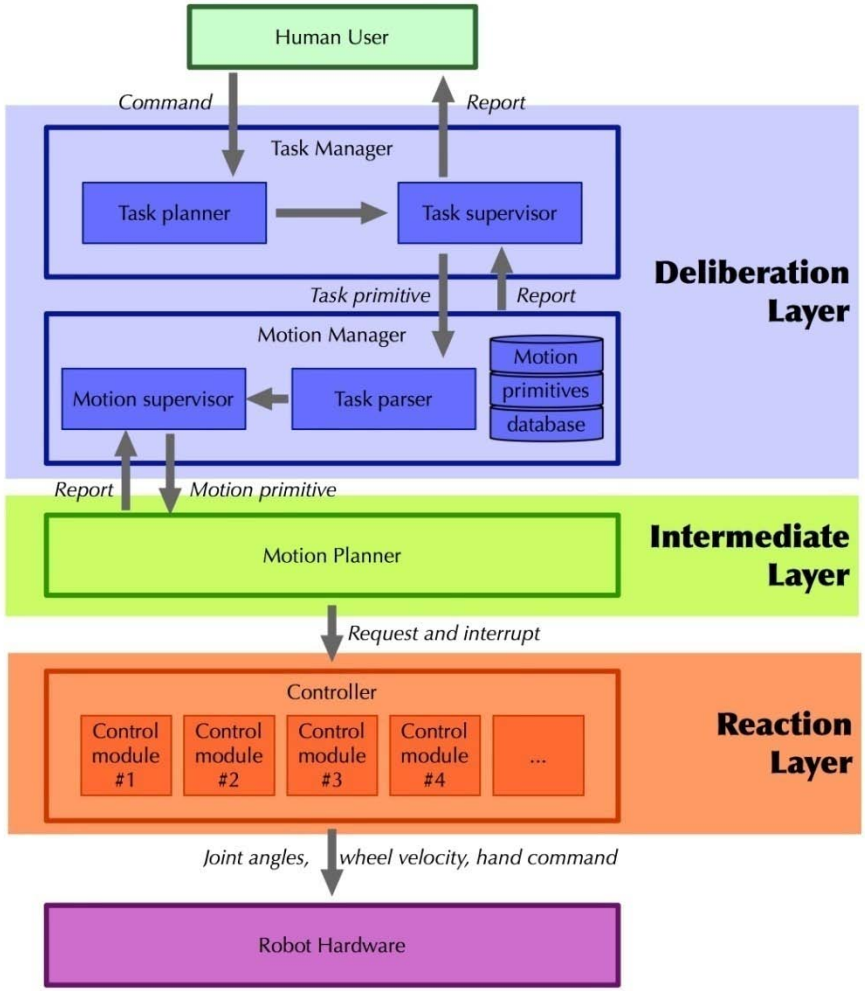
ROBOTICS LABORATORY

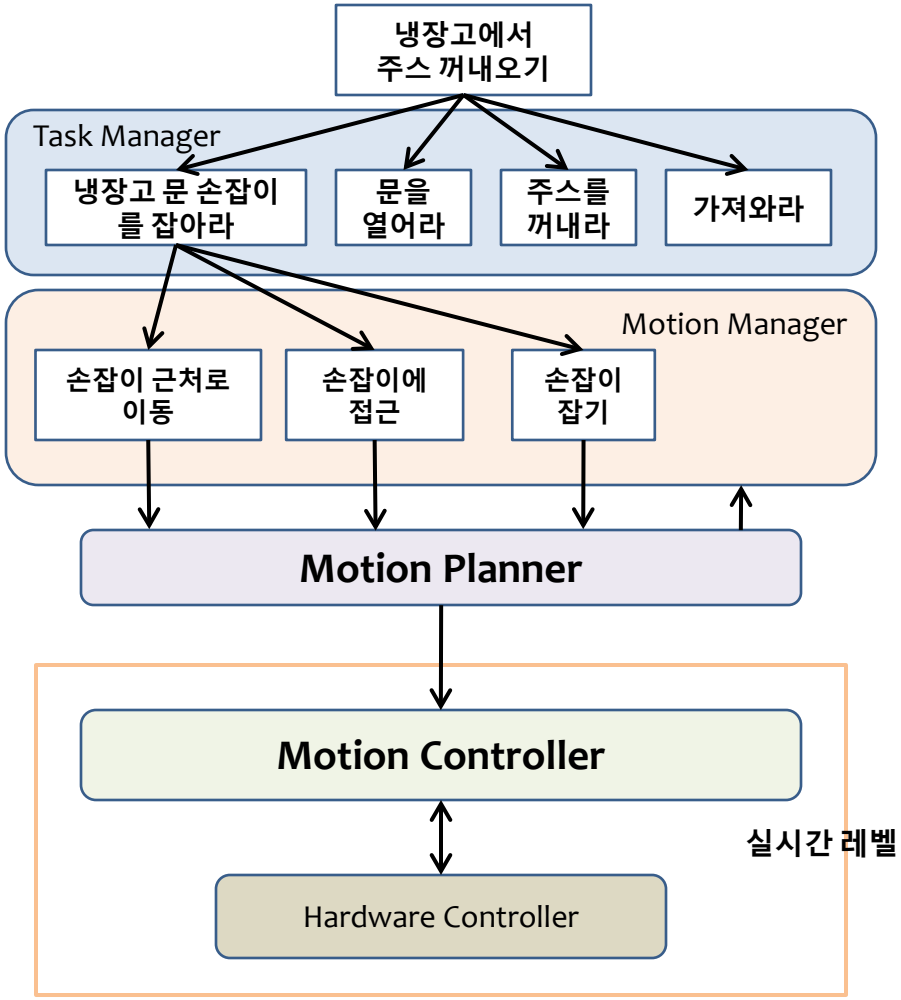
# Autonomous Robot Motion Planner

- Algorithms and software for motion planning and control of autonomous robot.



# The Scheme for Autonomous Motion Planning



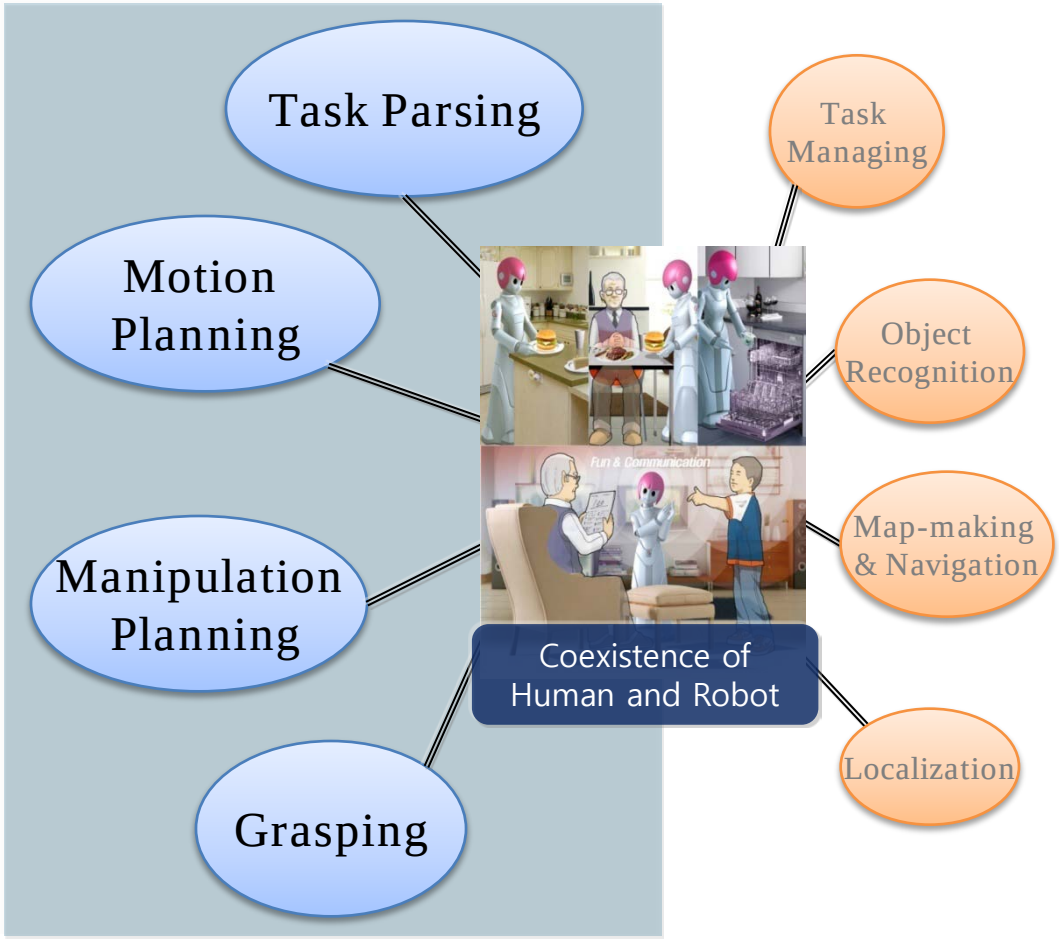
# Research Scope

- Task analysis
- Complex task segmentation
- Movement primitives
- Hierarchical command language

- Collision avoidance
- Finding feasible/optimal/natural motions
- Nonholonomic vehicles
- Legged robots
- Collaboration

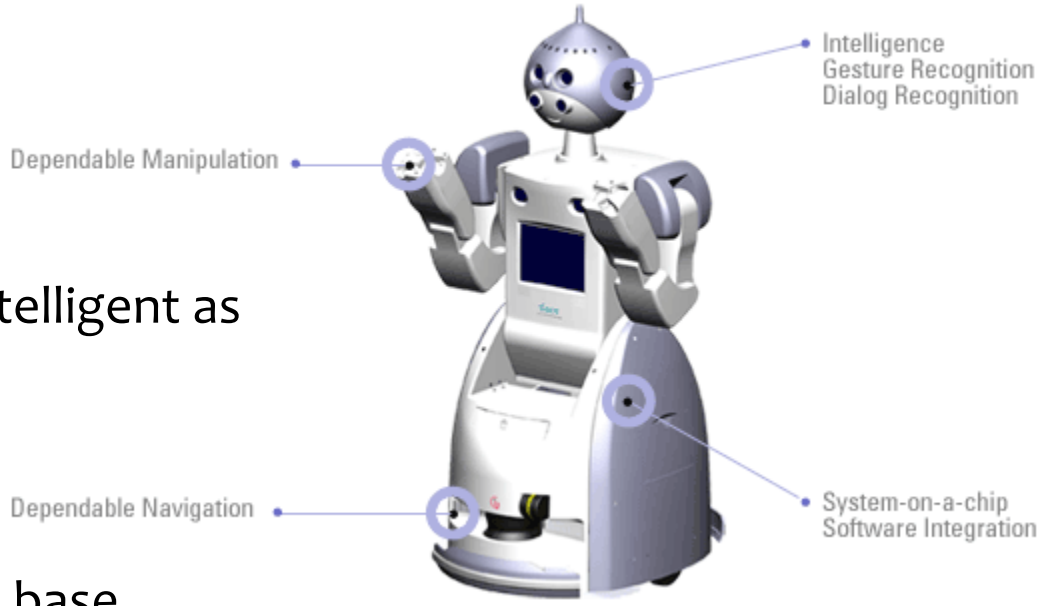
- Collision avoidance
- Finding feasible/optimal/natural motions
- Force/compliance planning & control
- Multi-arm manipulation/ closed chain control
- Multi-objective manipulation

- Primitive based grasp planning
- Stability measure of grasp
- Stable grasping using force sensor
- Grasp simulation



# Target Hardware: CIROS

- Service robot for the elder.
  - Dependable manipulation.
  - Dependable navigation.
  - Recognition, etc.
- At least, robot should be as intelligent as 5-year-old child.
- Two 7-DOF arms
- Two 3-fingered hands
- Two-differential-wheel mobile base. (nonholonomic system)

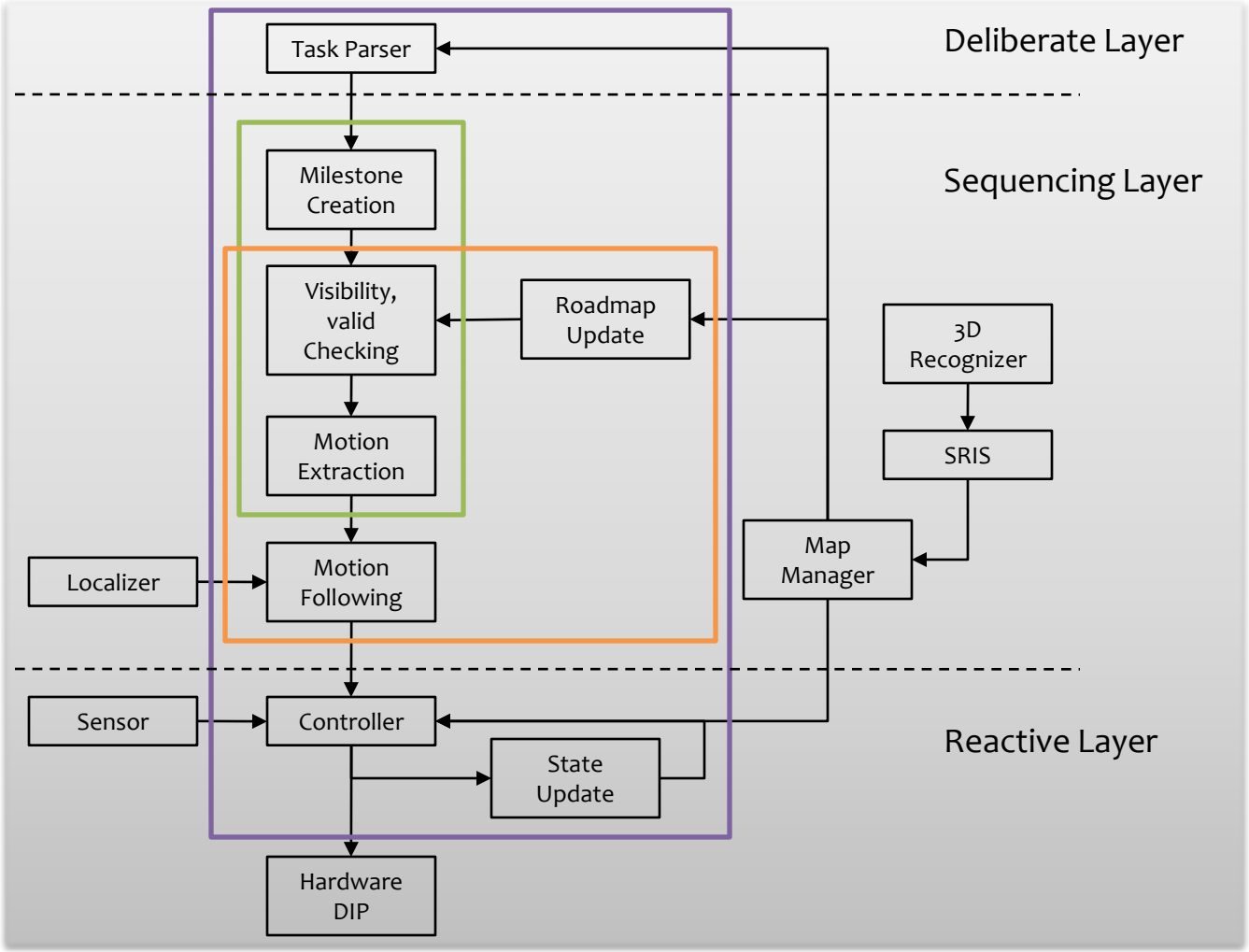



21C 프론티어사업 지능로봇사업단

KIST 한국과학기술연구원 Korea Institute of Science and Technology

산업자원부 21C 프론티어기술개발사업

# The Flow Chart of Motion Planning



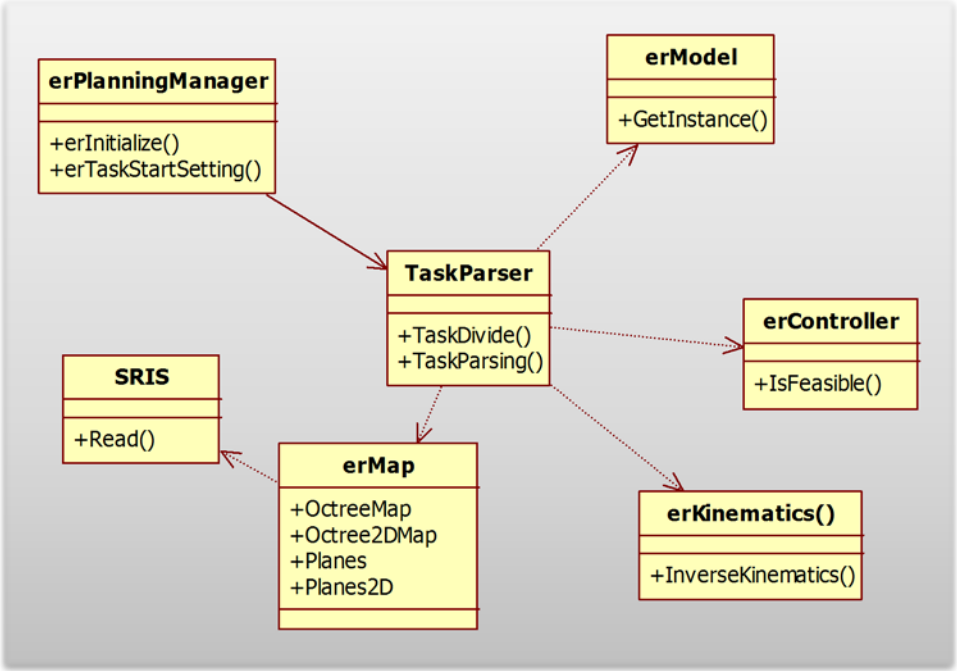

Motion Planner

Global Motion Planner

Local Motion Planner

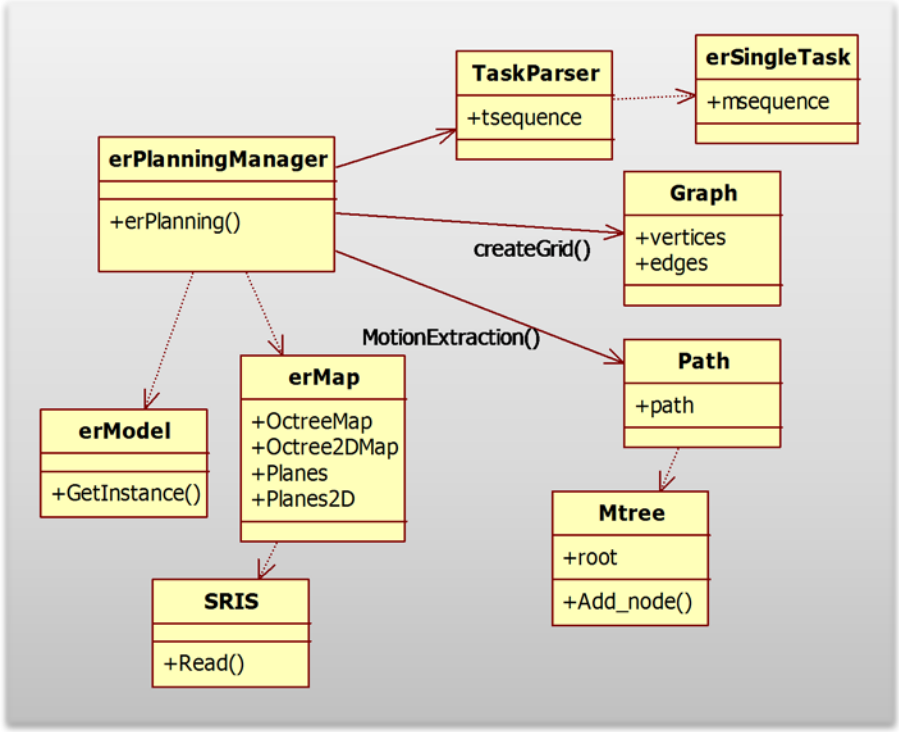
# Task Parser

- Task Parser
  - Read task commands from Task Manager.
  - A complex task is decomposed with subtasks.
  - A subtask is decomposed with motion primitives.
  - Set final configurations.
  - Build constraints.
  - Construct task environment.

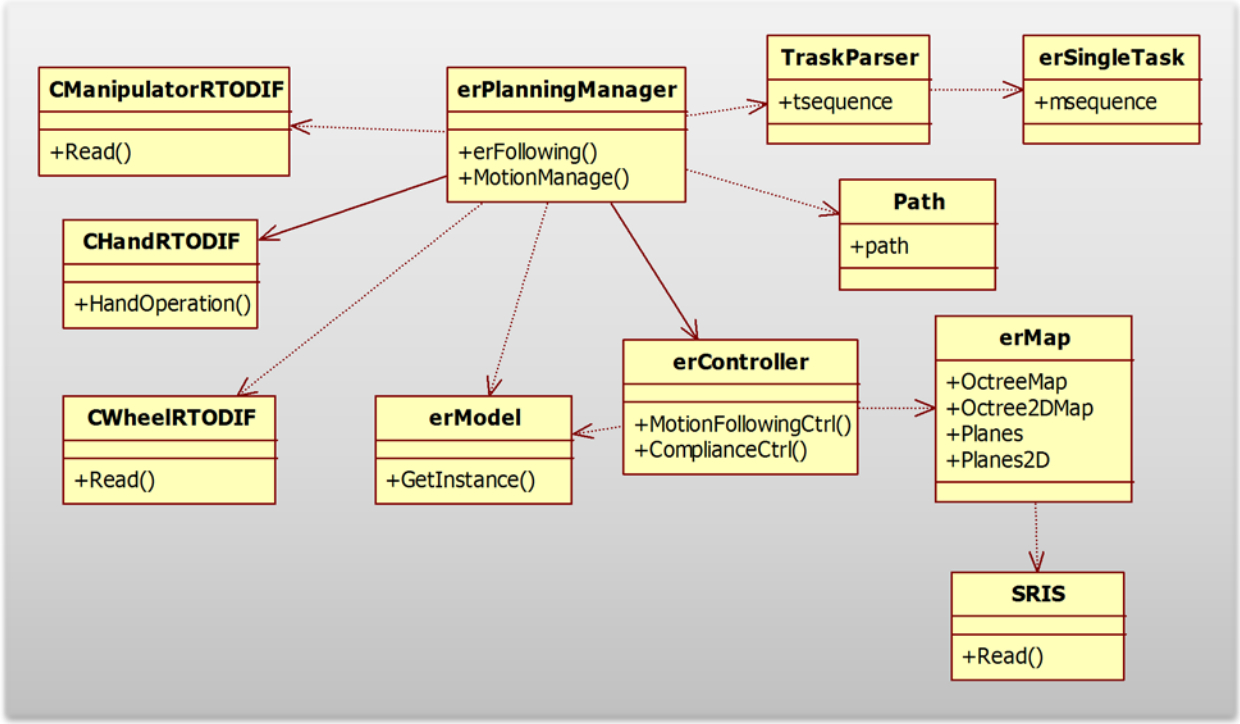


# Motion Planner

- Motion Planner
  - Global Motion Planner
    - Analyze predefined task space without moving obstacles.
    - Preprocess milestones for constructing collision-free paths.
    - Offline process.
  - Local Motion Planner
    - Find collision-free path with moving obstacles.
    - Online process.

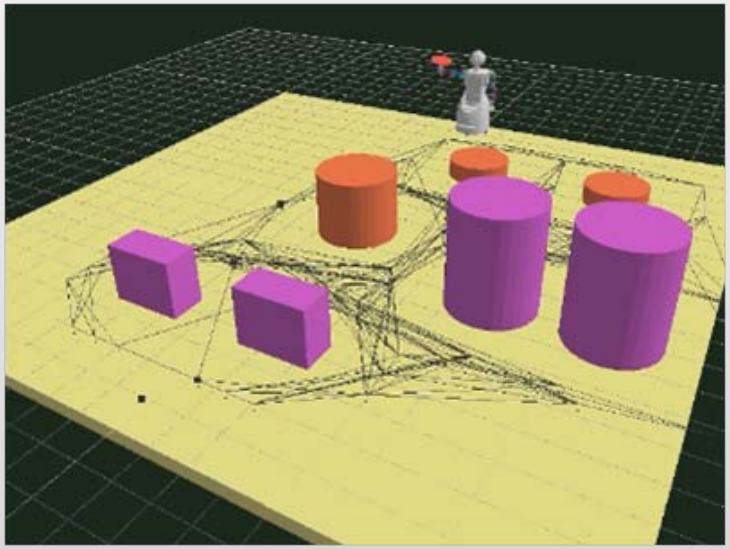


# Controller

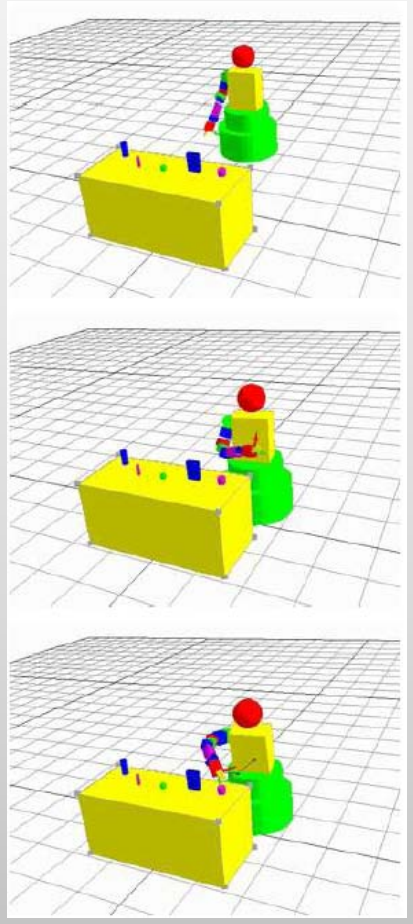



- Controller
  - Calculate control inputs from collision-free path which Motion Planner creates,
  - Considering given task and various constraints simultaneously,
  - and reading obstacle information from embedded sensors.

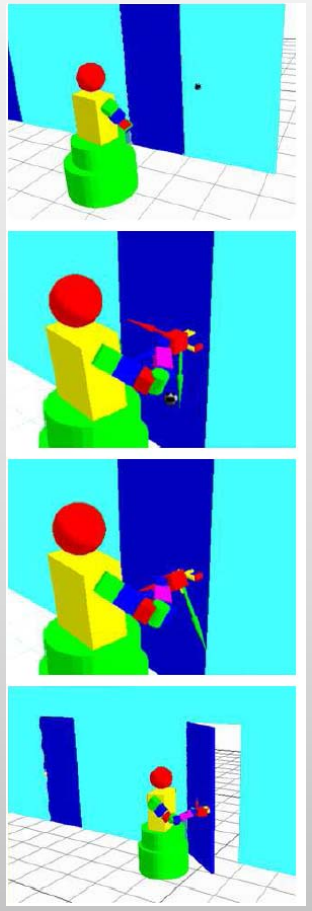
# Demonstrations



Collision-free
path planning

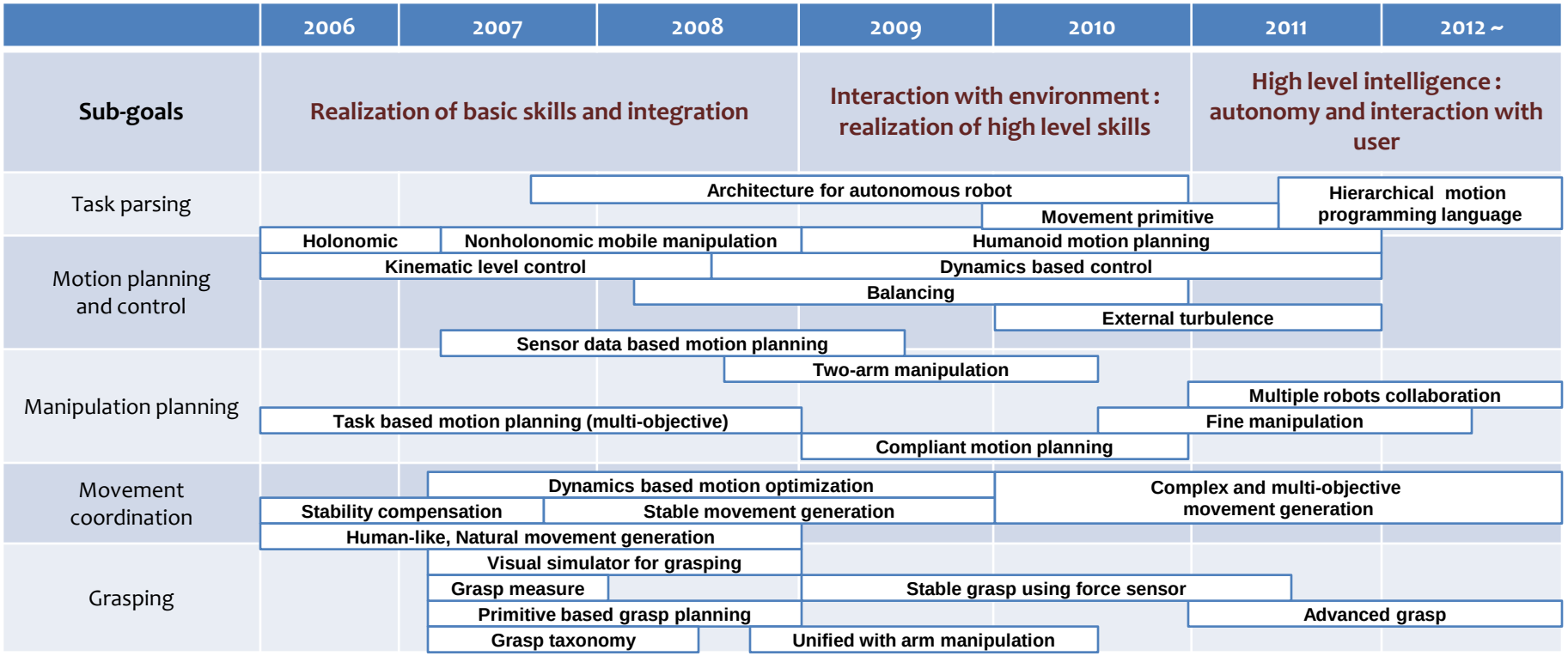


Grasping and object



Opening the door

# Technology Roadmap




- Motion Planner for Mobile Manipulator is suitable for software verification.

# Bugs to be considered

# Safety Bug & Security Bug

- Autonomous robot
  - is naturally unmanned mechanism.
  - always has a big red button for emergency.

- Bugs such as buffer overrun, and divide-by-zero, may cause disaster.
- Autonomous robot communicates wirelessly with a host computer or with coworkers.
  - working at hazardous environments.
  - multi-agent robotics.
  - military robots.
  - surveillance & security system.

# Functionality Bug

- Imprecise sensor data / Sensor data saturation.
- Temporary invalid sensor data.
- Improper control value.
  - cf. over/underflow
- Home position error.
- Wrong estimation of calculation time.
  - Real-time OS problem.
  - Many control modules work together.
  - Time sync is essential.


# Future Plan

# 연차별 목표

| 단 계 | 연도 | 목 표 |
|---|---|---|
| 1단계(4년)<br>로봇 소프트웨어<br>오류 검출기 | 1 | Motion planner for Mobile Manipulator 분석<br>Functionality bug 선정 |
| | 2 | Global/Local Motion Planner 모듈 및<br>Controller 모듈 오류 검출 기술 개발 |
| | 3 | Task Parser 모듈 오류 검출 기술 개발 |
| | 4 | 오류 검출 성능 개선 |