A Multi-Robot Platform for Mobil Robots with Multi-Agent Technology

Sebastian Rockel
8rockel@informatik.uni-hamburg.de

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

May 24, 2011

Outline

Present Situation

- complex tasks require specialized hardware (stereo vision, robot arm, robot hand..)
  - one »super bot« vs. multiple specialized robots
- augmented robot action radius (room, floor, area..)
  - need to cover a larger area in less time
- today's scenarios often need cooperation (e.g. kitchen scenario)
- lot of older and unused hardware in the lab
  - use available hardware for more complex tasks

Present Situation

- complex tasks require specialized hardware (stereo vision, robot arm, robot hand..)
  - one »super bot« vs. multiple specialized robots
- augmented robot action radius (room, floor, area..)
  - need to cover a larger area in less time
- today's scenarios often need cooperation (e.g. kitchen scenario)
- lot of older and unused hardware in the lab
  - use available hardware for more complex tasks

Present Situation

- complex tasks require specialized hardware (stereo vision, robot arm, robot hand..)
  - one »super bot« vs. multiple specialized robots
- augmented robot action radius (room, floor, area..)
  - need to cover a larger area in less time
- today's scenarios often need cooperation (e.g. kitchen scenario)
- lot of older and unused hardware in the lab
  - use available hardware for more complex tasks

Present Situation

- complex tasks require specialized hardware (stereo vision, robot arm, robot hand..)
  - one »super bot« vs. multiple specialized robots
- augmented robot action radius (room, floor, area..)
  - need to cover a larger area in less time
- today's scenarios often need cooperation (e.g. kitchen scenario)
- lot of older and unused hardware in the lab
  - use available hardware for more complex tasks

Why a Multi-Robot Platform?

Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Why a Multi-Robot Platform?

Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Why a Multi-Robot Platform?

Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Why a Multi-Robot Platform?

Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Why a Multi-Robot Platform?

Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Why a Multi-Robot Platform?

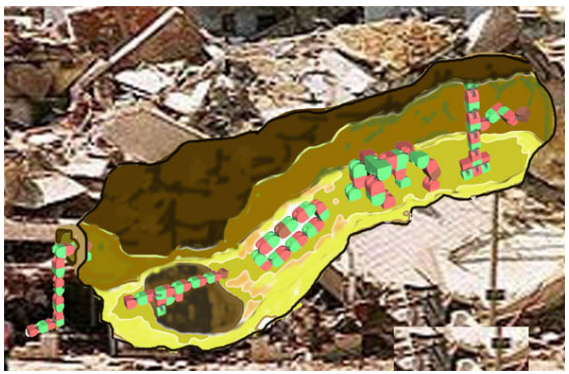
Advantages

- Complexity
  - multiple robots are capable to solve more complex tasks
- Performance
  - performance of multiple robots vs. single robot
- Flexibility
  - multiple robots can be assigned variety of different tasks
- Fault-tolerance
  - detection and replacement of malfunctioning robots
- Hardware re-use
  - older hardware, alone »useless«, can contribute to a team
- »Networking«
  - distributed and mobile sensing (sensor networks)

Research Topics

All about Cooperation

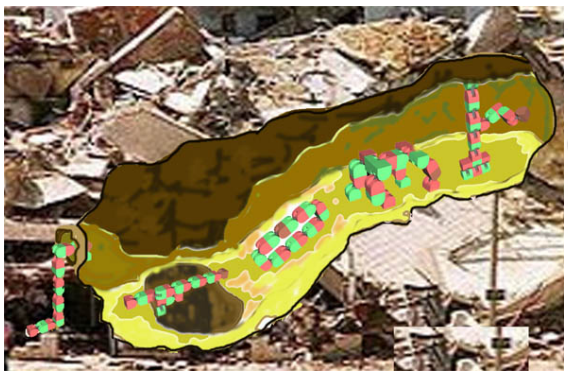
- search and rescue
- swarm formations
- exploration and SLAM
- inter-robot communication
- service robots



Research Topics

All about Cooperation

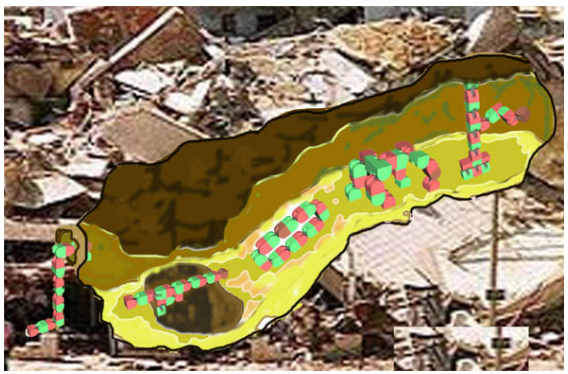
- search and rescue
- swarm formations
- exploration and SLAM
- inter-robot communication
- service robots

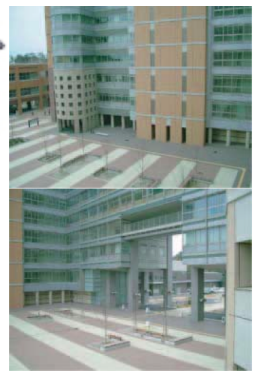
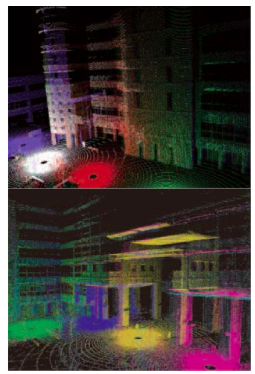


Research Topics

All about Cooperation

- search and rescue
- swarm formations
- exploration and SLAM
- inter-robot communication
- service robots

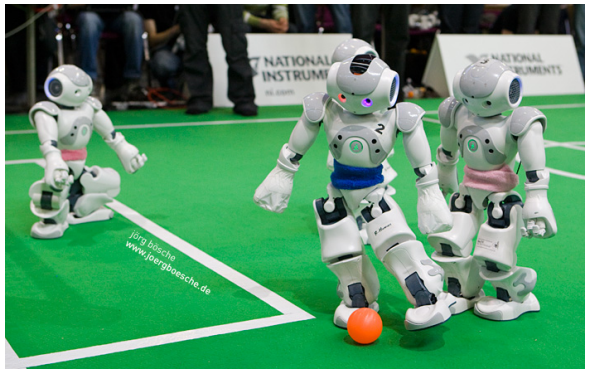


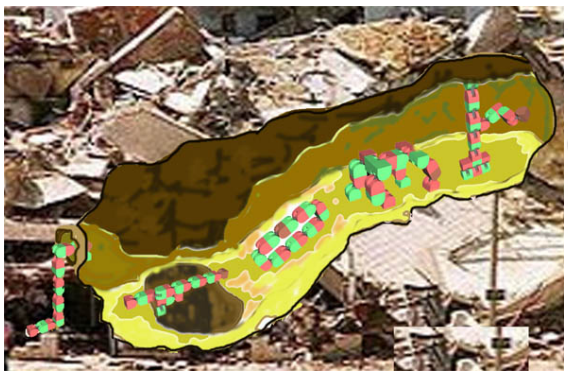



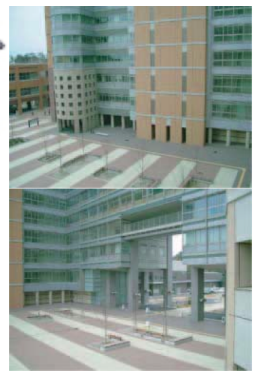
Research Topics

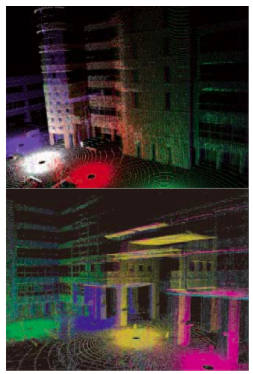
All about Cooperation

- search and rescue
- swarm formations
- exploration and SLAM
- inter-robot communication
- service robots





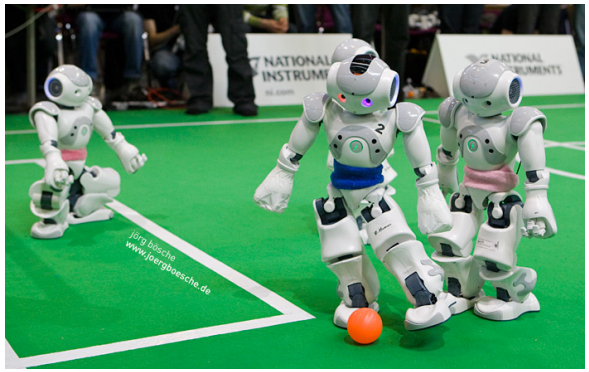


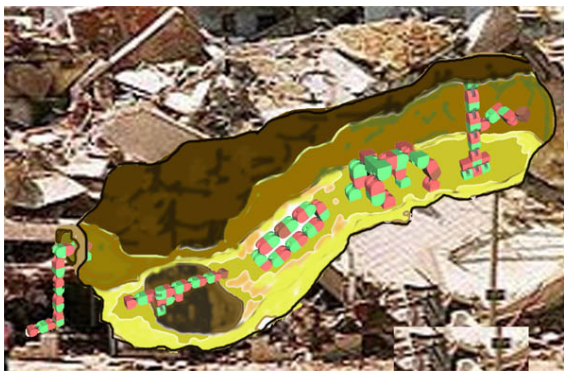


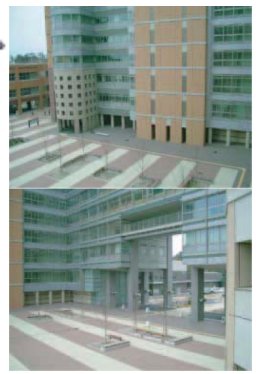
Research Topics

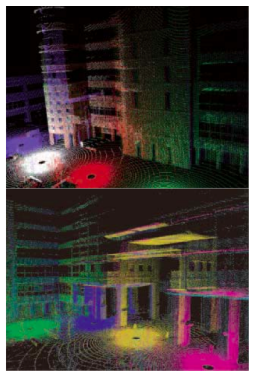
All about Cooperation

- search and rescue
- swarm formations
- exploration and SLAM
- inter-robot communication
- service robots





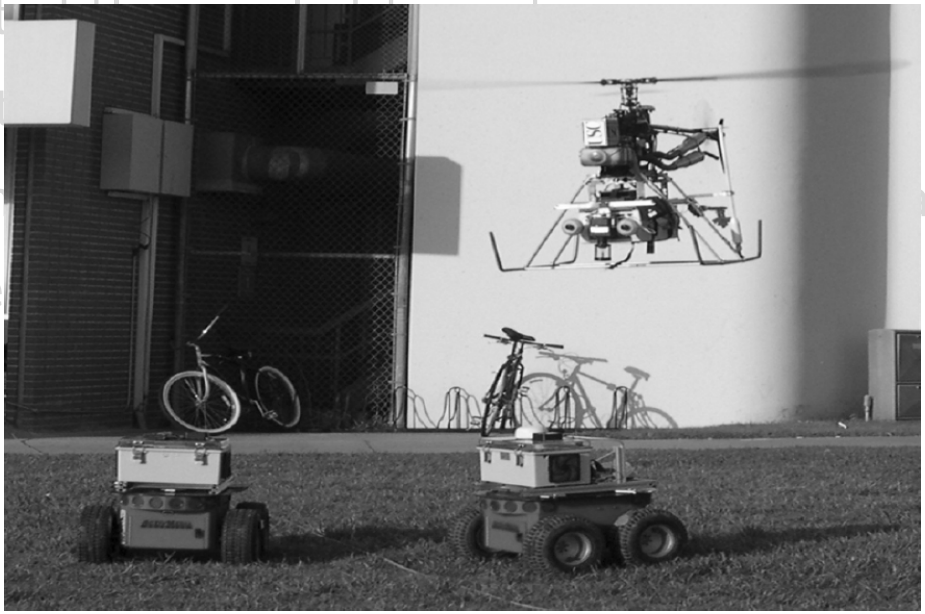




General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly coupled formation
- communication
- task assignment (localization)
- architecture
- dynamics in
- learning



General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

General Issues [Bek05]

Multi-Robot Platform

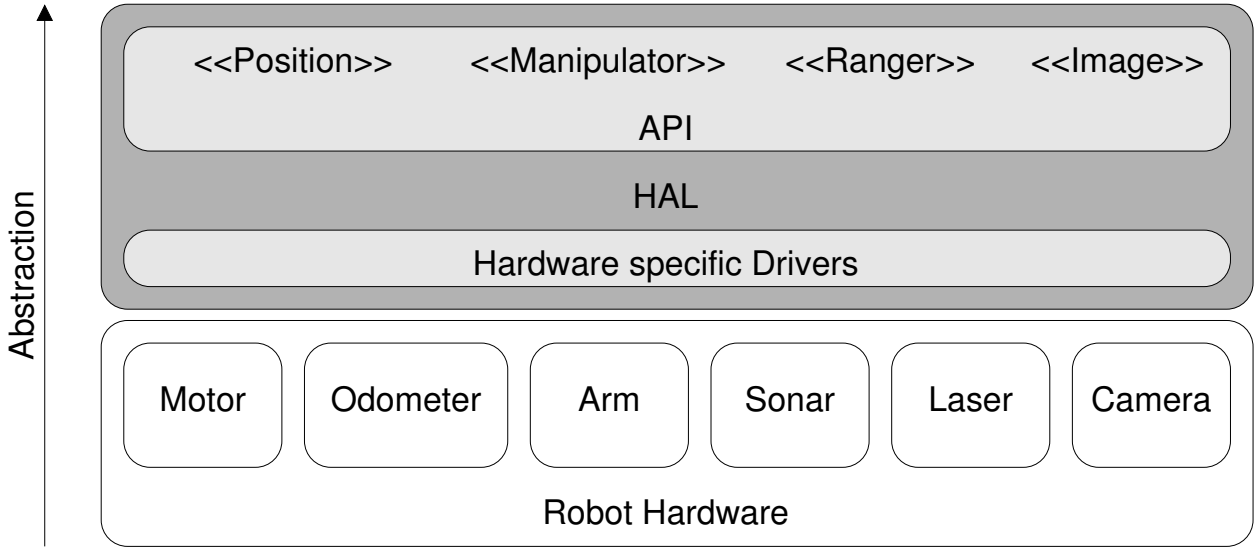
- homogeneous vs. heterogeneous robots
- centralized vs. distributed control
- loosely vs. tightly-coupled formations
- communication (broadcast, point to point)
- task assignment (negotiation, blackboard, specialization)
- architectures
- dynamics in environment, hardware, tasks
- learning

Outline

Multi-Robot System (MRS)

Features

- hardware abstraction layer (HAL, driver for hardware)
- algorithms for robot control (localization, navigation)
- Network layer (peer-to-peer, socket) (4)
- »robot operating system«



Multi-Robot System (MRS)

Features

- hardware abstraction layer (HAL, driver for hardware)
- algorithms for robot control (localization, navigation)
- Network layer (peer-to-peer, socket) (4)
- »robot operating system«

Multi-Robot System (MRS)

Features

- hardware abstraction layer (HAL, driver for hardware)
- algorithms for robot control (localization, navigation)
- Network layer (peer-to-peer, socket) (4)
- »robot operating system«

Multi-Robot System (MRS)

Features

- hardware abstraction layer (HAL, driver for hardware)
- algorithms for robot control (localization, navigation)
- Network layer (peer-to-peer, socket) (4)
- »robot operating system«

MRS Overview

- Player/Stage
- Robot Operating System (ROS)
- similar features and algorithms (navigation, SLAM)
- different programming languages (Java)
- others

Note:
»Multi« here does not mean cooperation

MRS Overview

- Player/Stage
- Robot Operating System (ROS)
- similar features and algorithms (navigation, SLAM)
- different programming languages (Java)
- others

Note:

»Multi« here does not mean cooperation

Player/Stage

- since 1999[1]
- C, C++, Ada, Octave and Java, as well as Ruby and Python
- Simulator Stage (2.5D, performance), Gazebo (3D, high fidelity)
- out-of-the-box tools included for navigation and visualization sensor data
- Server/Client architecture
- Javaclient[2]
- Socket interface

[1] http://playerstage.sourceforge.net
[2] http://java-player.sourceforge.net

ROS

Robot Operating System

- similar features
- used Player/Stage code and others
- used Player/Stage simulator Stage
- does currently not natively support Java (alpha state JNI client[3] though), but C++, Python, Octave, and LISP
- Peer-to-peer network message interface via XML-RPC

[3] http://www.ros.org/wiki/rosjava

Summary

Pro Player/Stage

- client and server program can run on different machines (sockets)
- well integrated Stage and Gazebo simulators
- many integrated algorithms available (AMCL, VFH, ICP, SLAM)
- native, well-supported, functional Java-Interface
- preliminary work experience [RG10]

Summary

Pro Player/Stage

- client and server program can run on different machines (sockets)
- well integrated Stage and Gazebo simulators
- many integrated algorithms available (AMCL, VFH, ICP, SLAM)
- native, well-supported, functional Java-Interface
- preliminary work experience [RG10]

Summary

Pro Player/Stage

- client and server program can run on different machines (sockets)
- well integrated Stage and Gazebo simulators
- many integrated algorithms available (AMCL, VFH, ICP, SLAM)
- native, well-supported, functional Java-Interface
- preliminary work experience [RG10]

Summary

Pro Player/Stage

- client and server program can run on different machines (sockets)
- well integrated Stage and Gazebo simulators
- many integrated algorithms available (AMCL, VFH, ICP, SLAM)
- native, well-supported, functional Java-Interface
- preliminary work experience [RG10]

Summary

Pro Player/Stage

- client and server program can run on different machines (sockets)
- well integrated Stage and Gazebo simulators
- many integrated algorithms available (AMCL, VFH, ICP, SLAM)
- native, well-supported, functional Java-Interface
- preliminary work experience [RG10]

Outline

Multi-Agent System (MAS)

- agent paradigm
  - autonomous object
  - dynamic (active)
- agent platform provides distribution (over network)
- many platforms available, to be picked by feature,license, included tools

perfect:

pick up agents needed for a task and they would re-organize themselves to reach goal

Multi-Agent System (MAS)

- agent paradigm
  - autonomous object
  - dynamic (active)
- agent platform provides distribution (over network)
- many platforms available, to be picked by feature,license, included tools

perfect:

pick up agents needed for a task and they would re-organize themselves to reach goal

Jadex (JADE eXtension)

- since ca. 2003 (VSIS)
- originally extension to JADE (Java Agent DEvelopment framework), standalone platform available today
- middleware and reasoning feature
  - network layer and goal-orientation (BDI-model)
- FIPA compliance[4]
- agent debugger
- platform independent: Java and XML
- optimized for distribution and concurrency

[4] http://www.fipa.org

Outline

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Related Work

Assorted Examples

- navigation (with JADE)
  - Negotiation protocol for shortest distance [Cer08]
  - Navigation in unknown territory [SN10]
- lot of formation research, e.g. dynamic approach [STSI09]
- building a map (3D) with a team of heterogenous robots [KNT+09]
- task assignment and subdivision [SC09]
- software design, re-usable [TvS10]
- learning, reinforcement learning in MRS [SMKR09]
- Mixed Reality [CMW09]

Issues

- typically proprietary (MRS/MAS) systems
- no modularity
- hardware-specific
- no transparent network layer

Issues

- typically proprietary (MRS/MAS) systems
- no modularity
- hardware-specific
- no transparent network layer

Issues

- typically proprietary (MRS/MAS) systems
- no modularity
- hardware-specific
- no transparent network layer

Issues

- typically proprietary (MRS/MAS) systems
- no modularity
- hardware-specific
- no transparent network layer

Outline

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Goals

- use existing MRS and MAS
- new middle layer connecting MAS and MRS (3 layer approach)
  - »Robot System Abstraction Layer« (RSAL)
- abstract from MRS and encapsulate agent code
- flexibility: exchange either MAS, MRS or both
- de-couple MAS and MRS
- provides asynchronous API
- performance optimization

Outline

Architecture

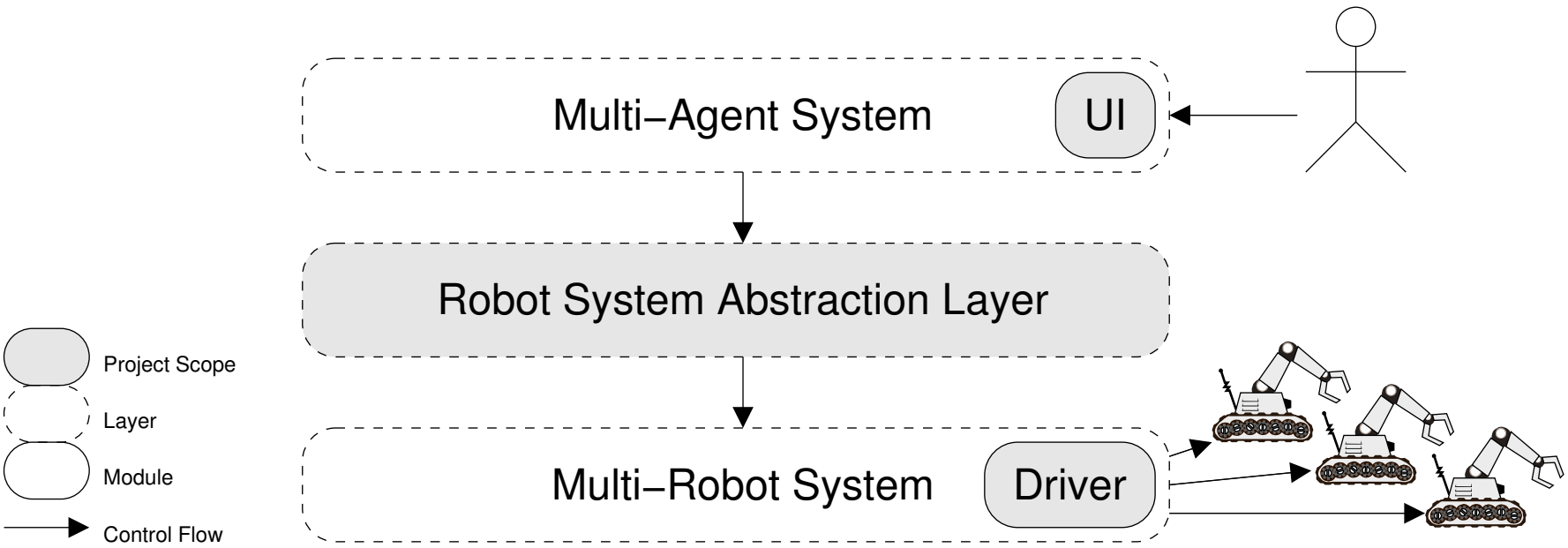


Figure: 3 Layer Architecture

Architecture (cont.)

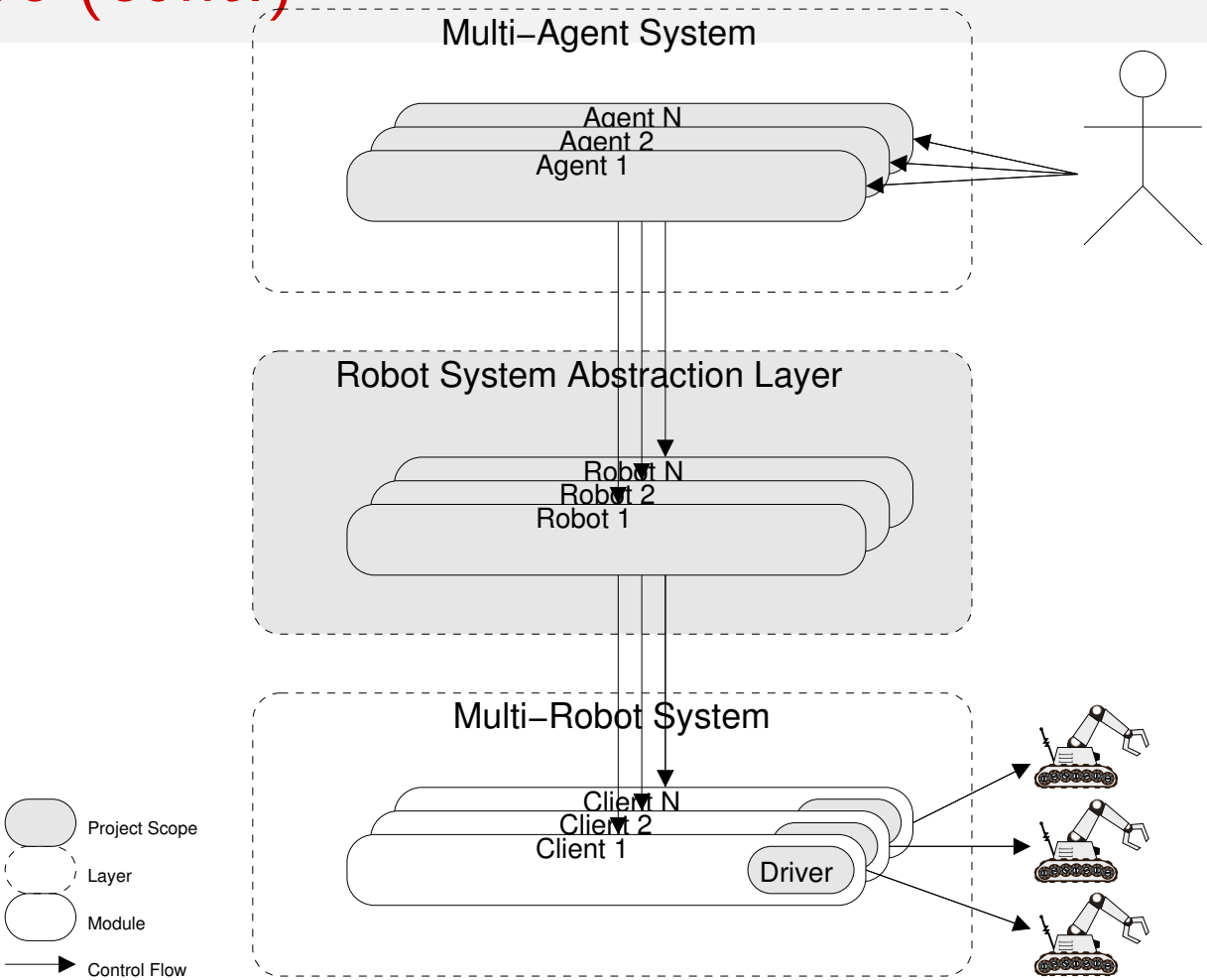


Figure: 3 Layer Architecture, detailed

Outline

Design

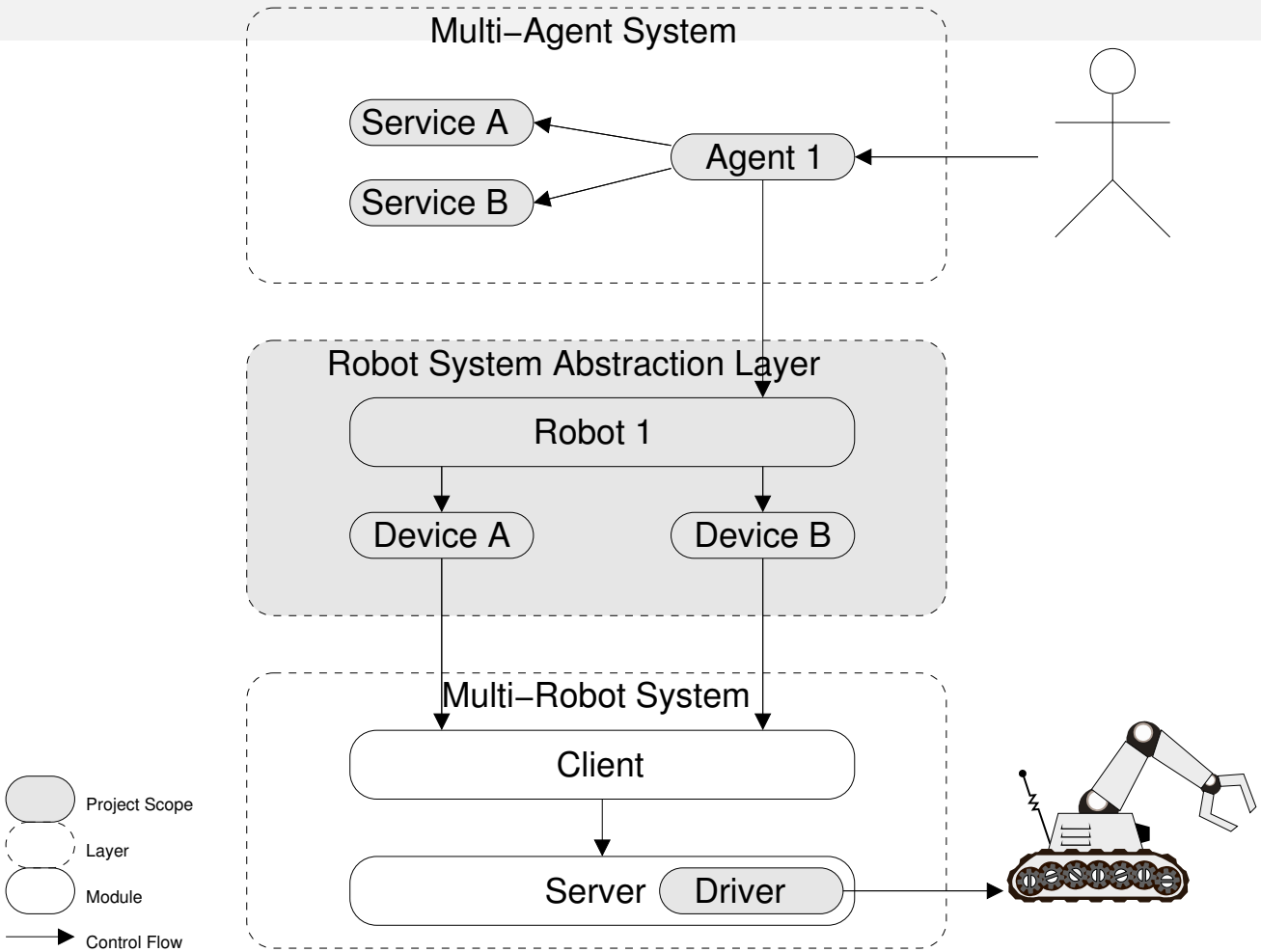


Figure: One Agent Example

Design (cont.)

Services

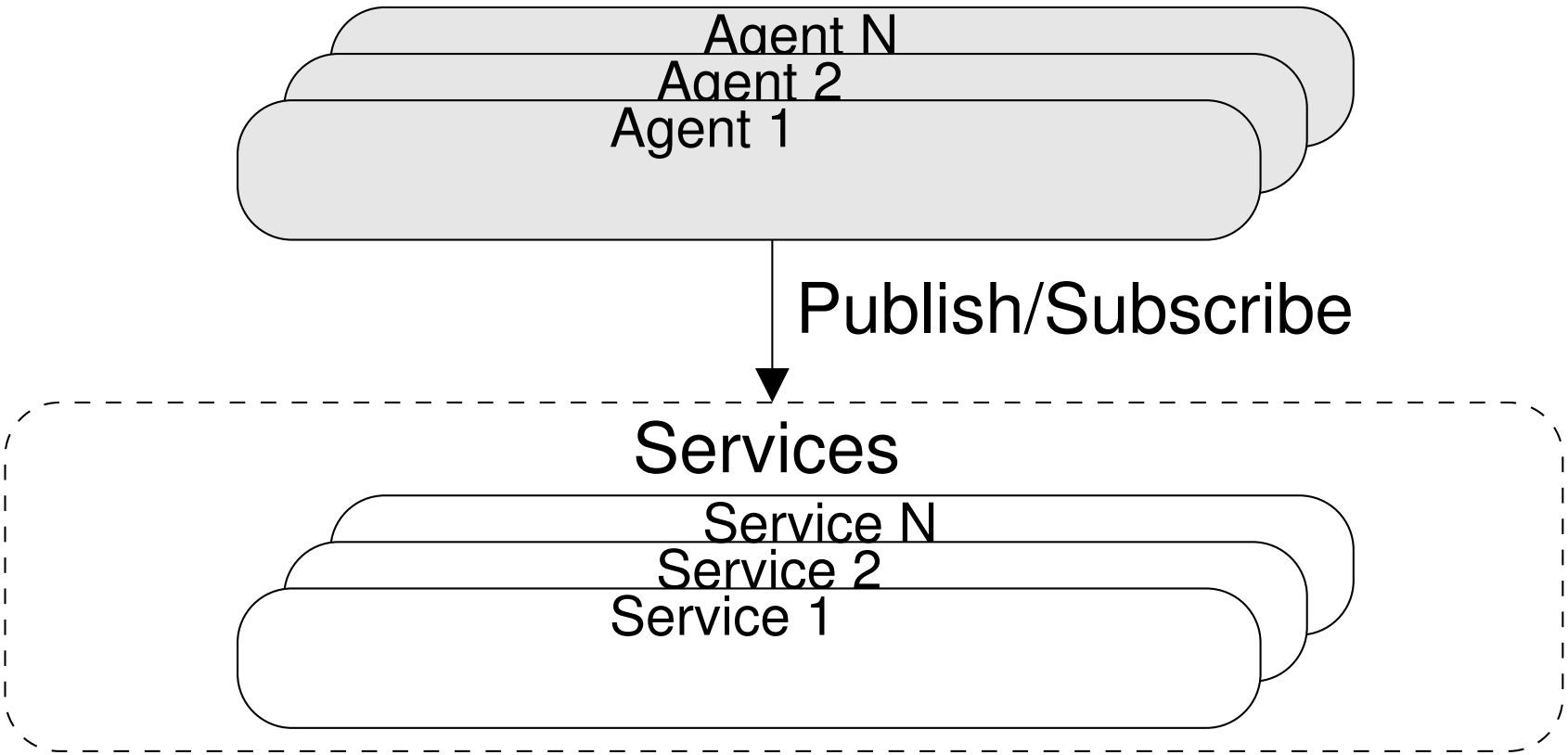


Figure: Agent Services

Design (cont.)

Communication

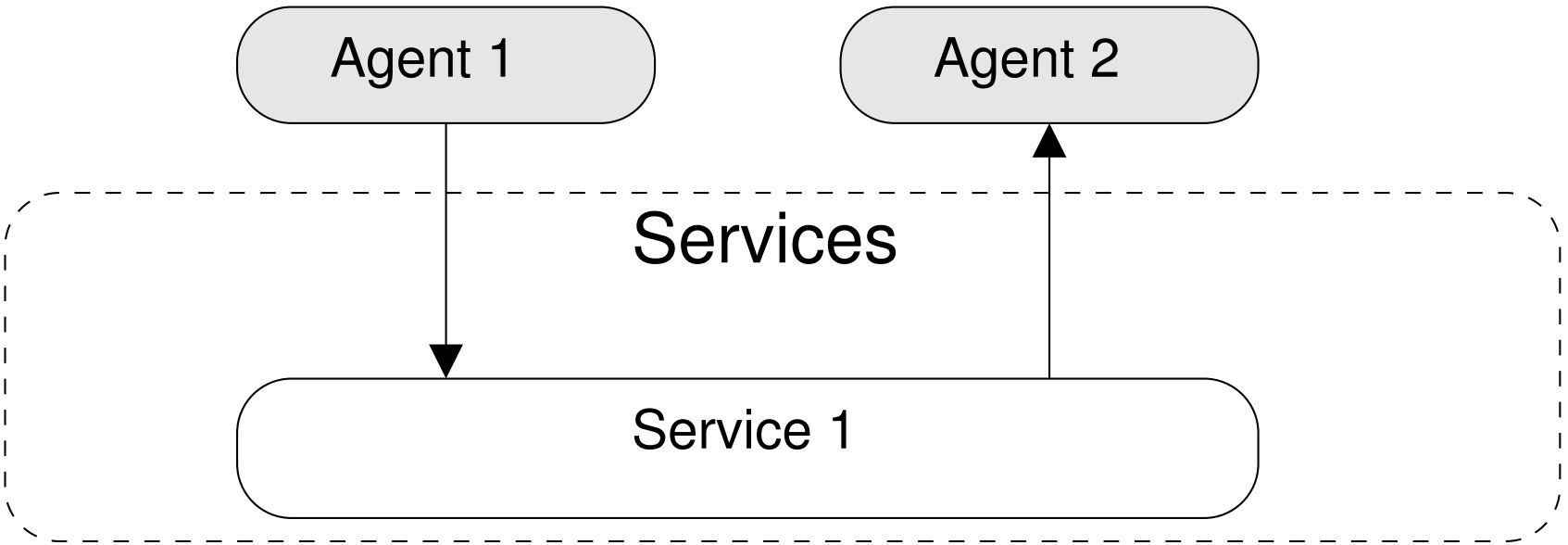
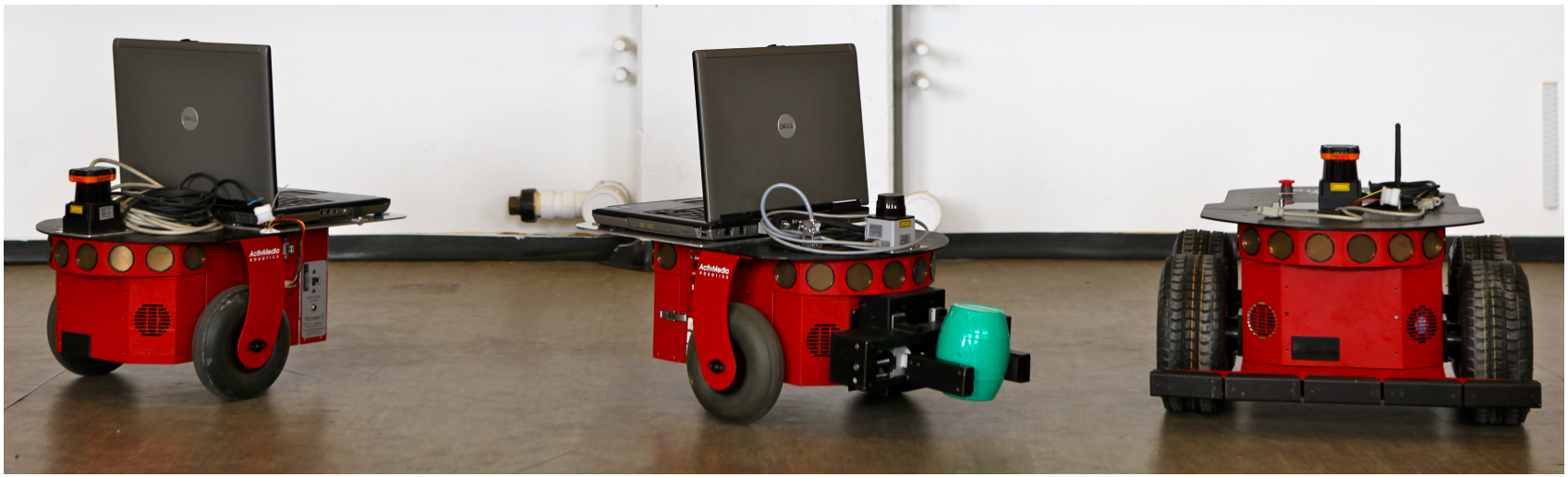


Figure: Agent Communication

Outline

Current Hardware

- 2 x Pioneer 2-DX
- 1 x Pioneer 3-AT
  - motor, odometer, gripper, 8 and 16 sonar ranger sensors per robot
- mounted laser ranger finder Hokuyo UTM-30LX (ca. 60m) and URG-04LX (ca. 5m)



Map Creation

- localization relies on accurate map
- otherwise unpredictable robot behavior while navigating

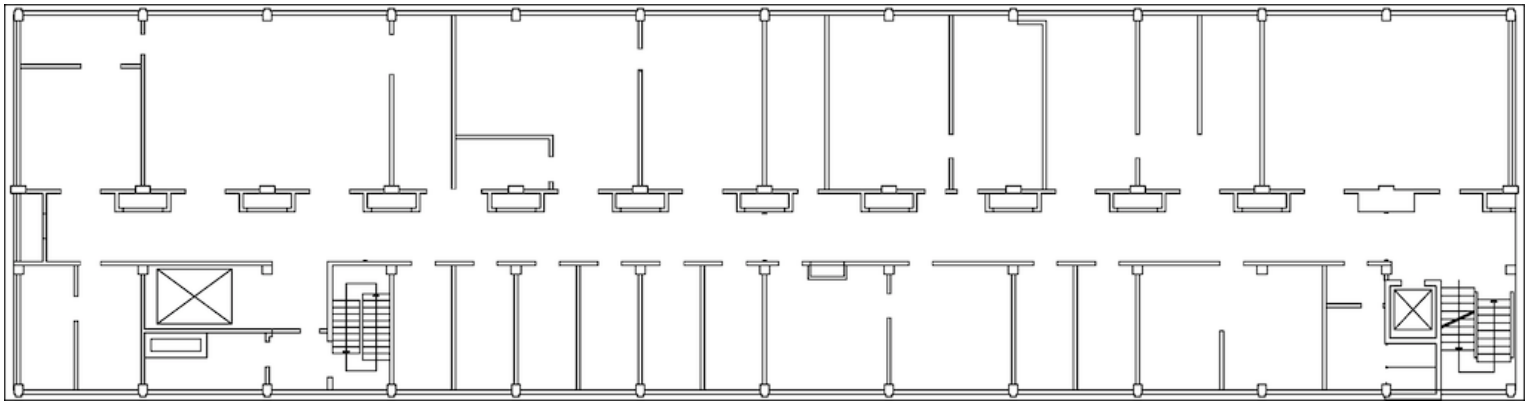


Figure: Original topological map

Map Creation

- localization relies on accurate map
- otherwise unpredictable robot behavior while navigating

Figure: Grid-map created with Player/Stage on a Pioneer 3-AT with a Hokuyo UTM-30LX laser ranger (~60 meters) attached

Map Creation

- localization relies on accurate map
- otherwise unpredictable robot behavior while navigating

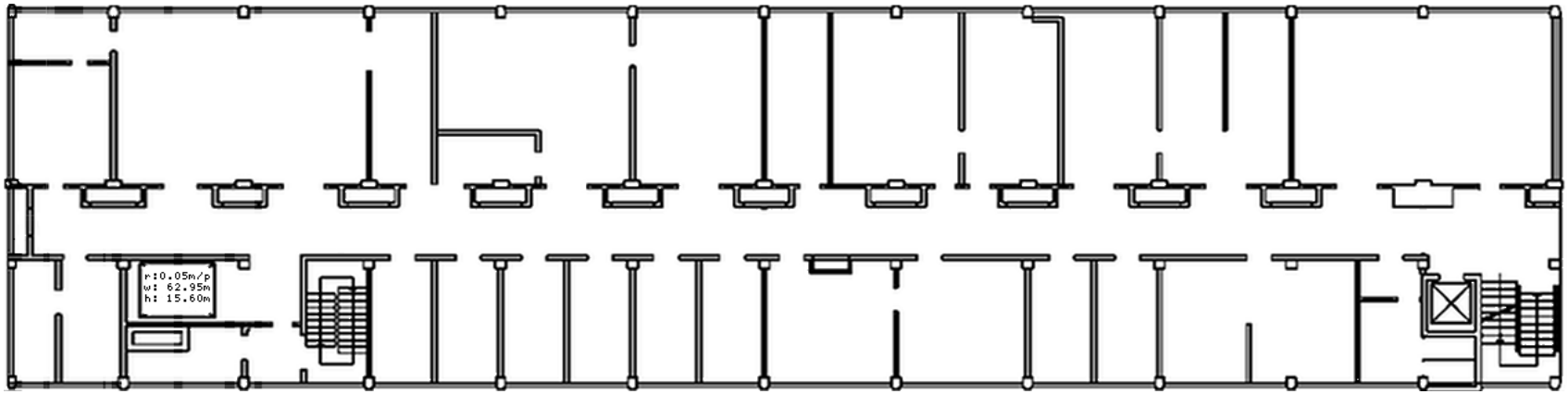


Figure: Composed accurate map (5 cm per pixel)

Map Creation

- localization relies on accurate map
- otherwise unpredictable robot behavior while navigating

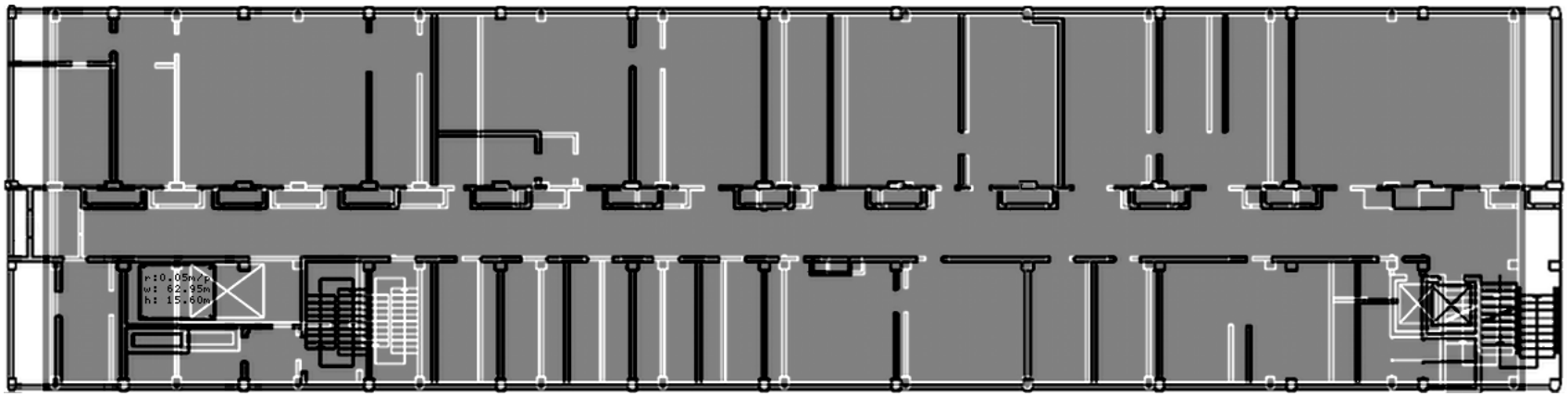


Figure: Error between accurate and original map

MRS Drivers

Player/Stage

- global path planner
  - Wavefront
- local path planner
  - Vector Field Histogram (VHF)
  - Nearness Diagram (ND)
  - Smoothed Nearness Diagram (SND)
- localization
  - Adaptive Monte Carlo Localization (AMCL)

Outline

Demo Scenarios

- Hunter and Prey (today)
  - prey and hunter robots
- Cleaner World
  - search and collect robot
- Swarm Distribution
  - with a global planner
- 100 Robots
  - scalability

Mixed Reality

- last one virtual only
- others can run real, virtual or mixed

Hunter and Prey

Scenario Definition

- two robots hunt one prey robot
- components: Escape Agent and Follow Agent

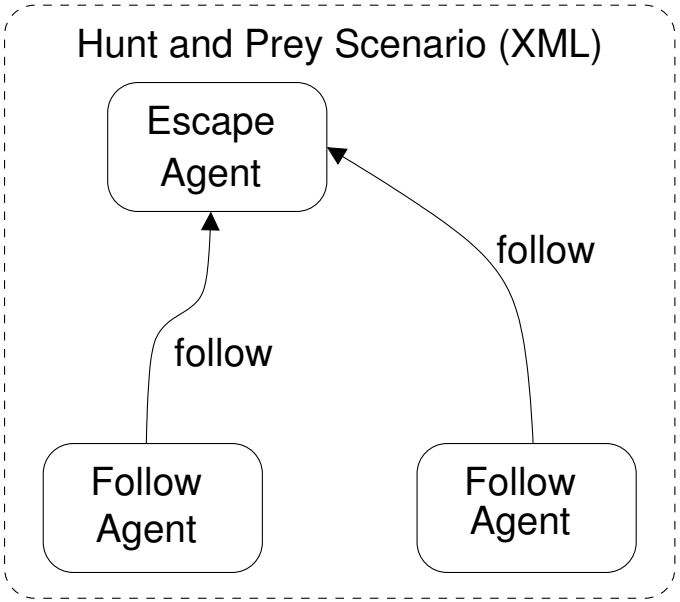


Figure: Hunt and Prey Scenario

Hunter and Prey (cont.)

Escape Agent Design

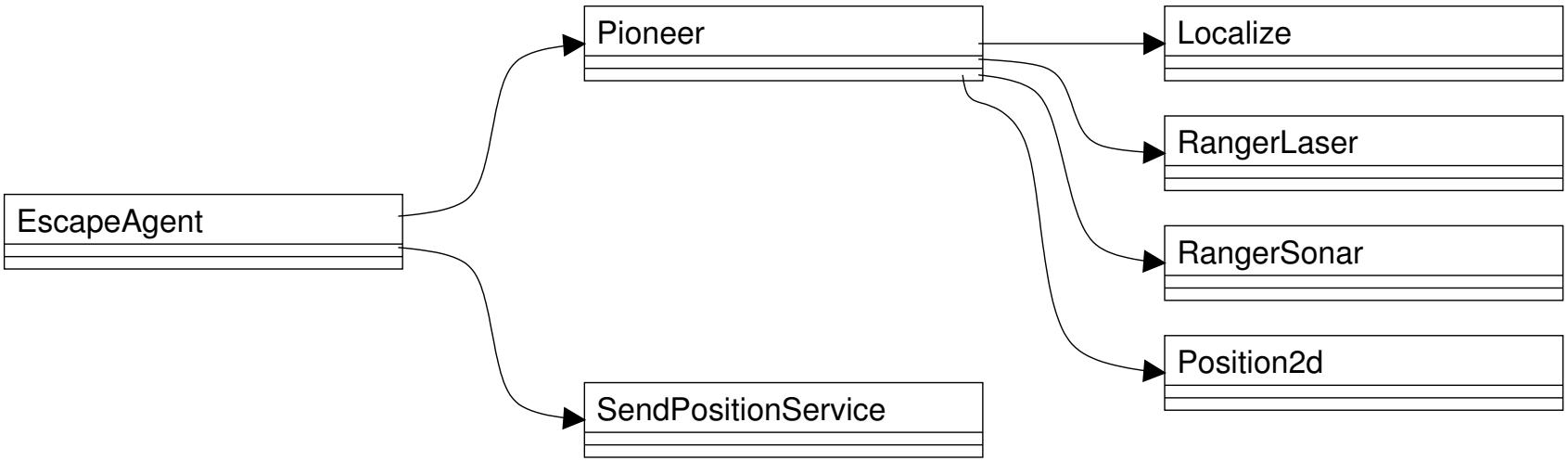


Figure: Escape Agent, »has-a« relation

Hunter and Prey (cont.)

Follow Agent Design

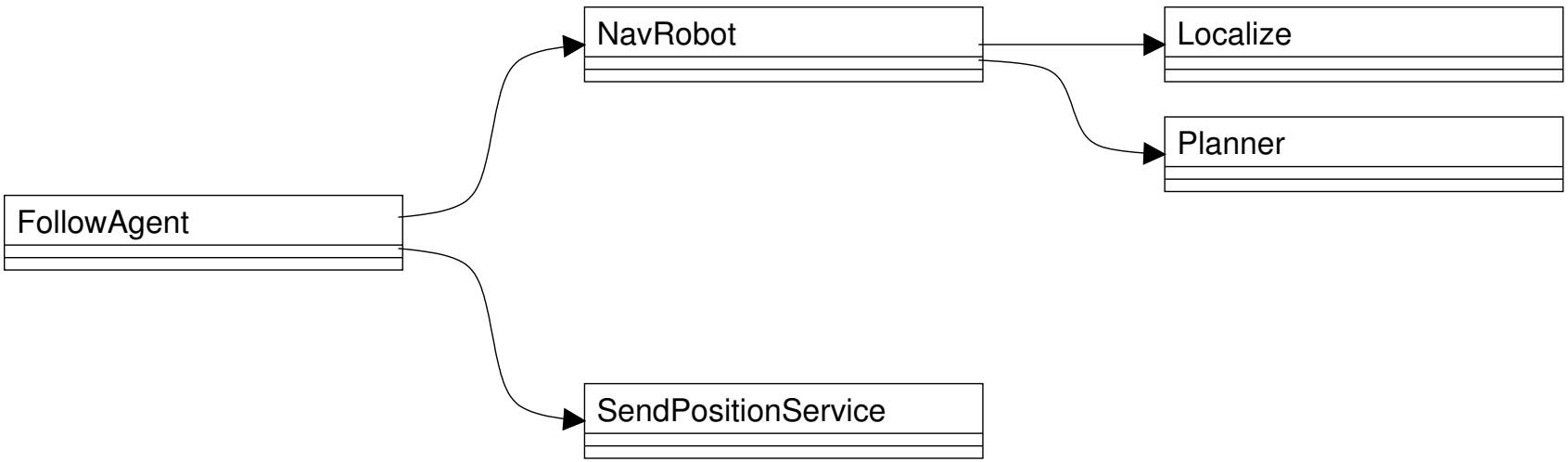


Figure: Follow Agent, »has-a« relation

Hunter and Prey (cont.)

Scenario Dynamic View

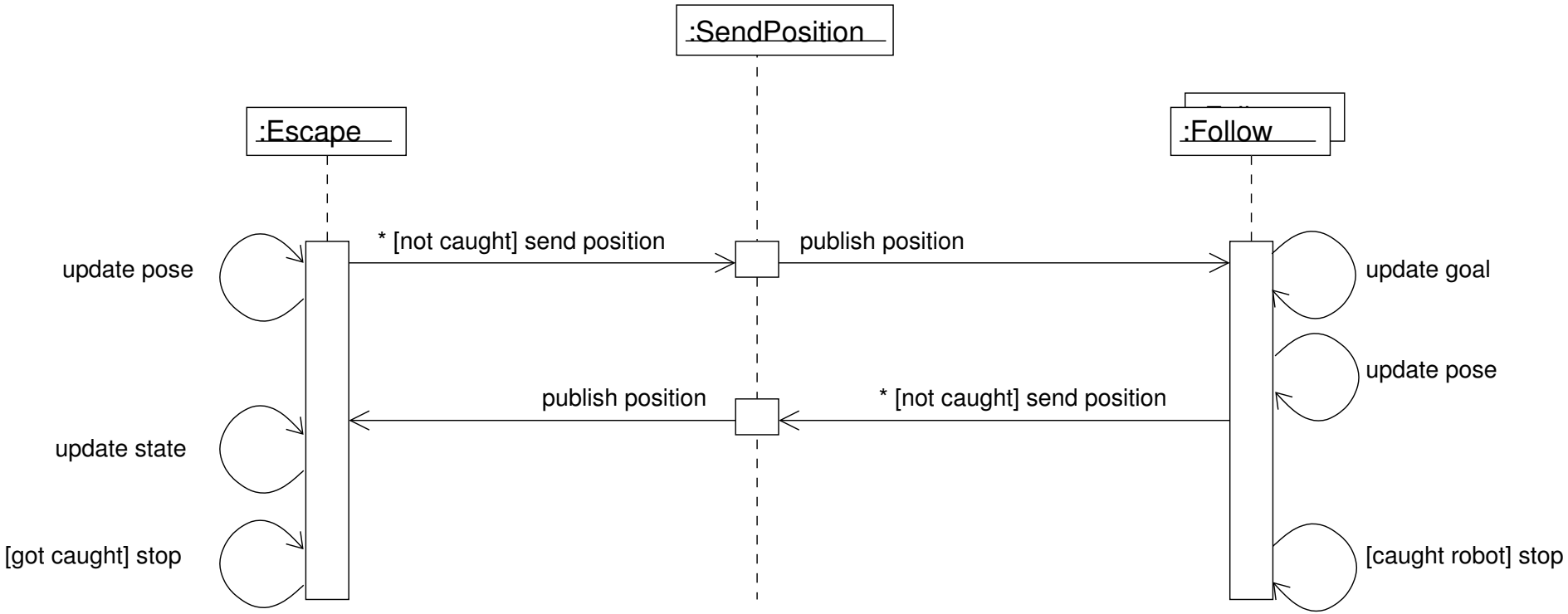


Figure: Hunt and Prey sequence (omitted initialization and registration)

Hunter and Prey (cont.)

Pioneer Robot Behavior Architecture

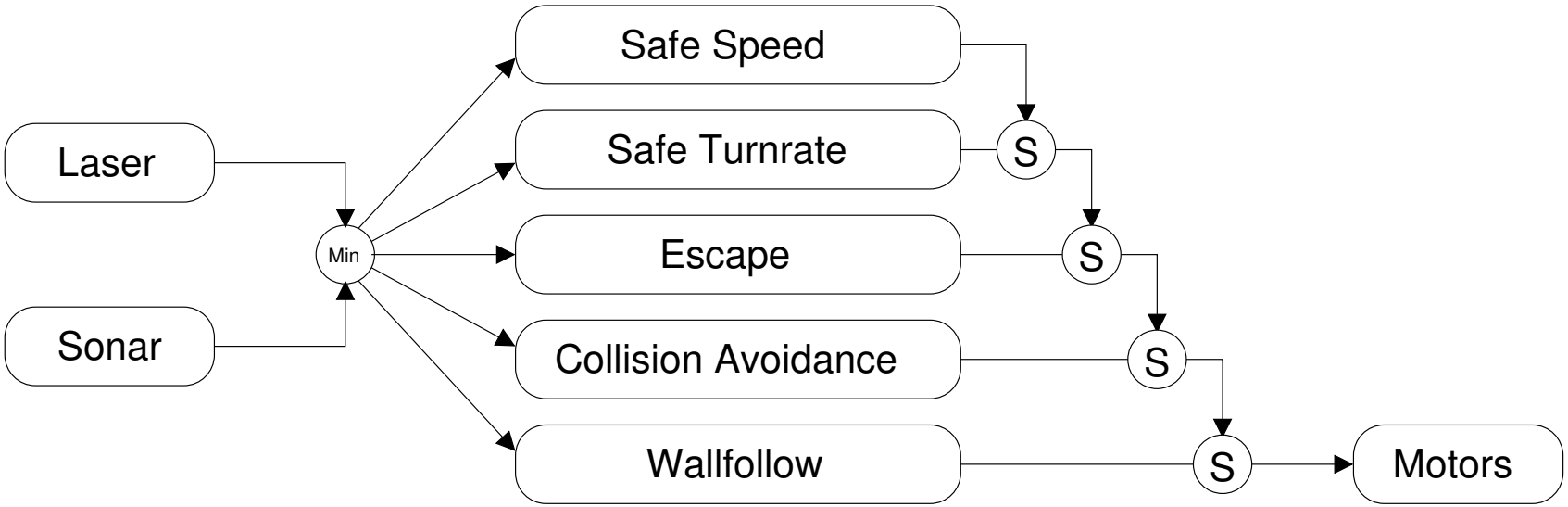


Figure: Pioneer Behavior (subsumption model)

Hunter and Prey (cont.)

Agent Class Hierarchy

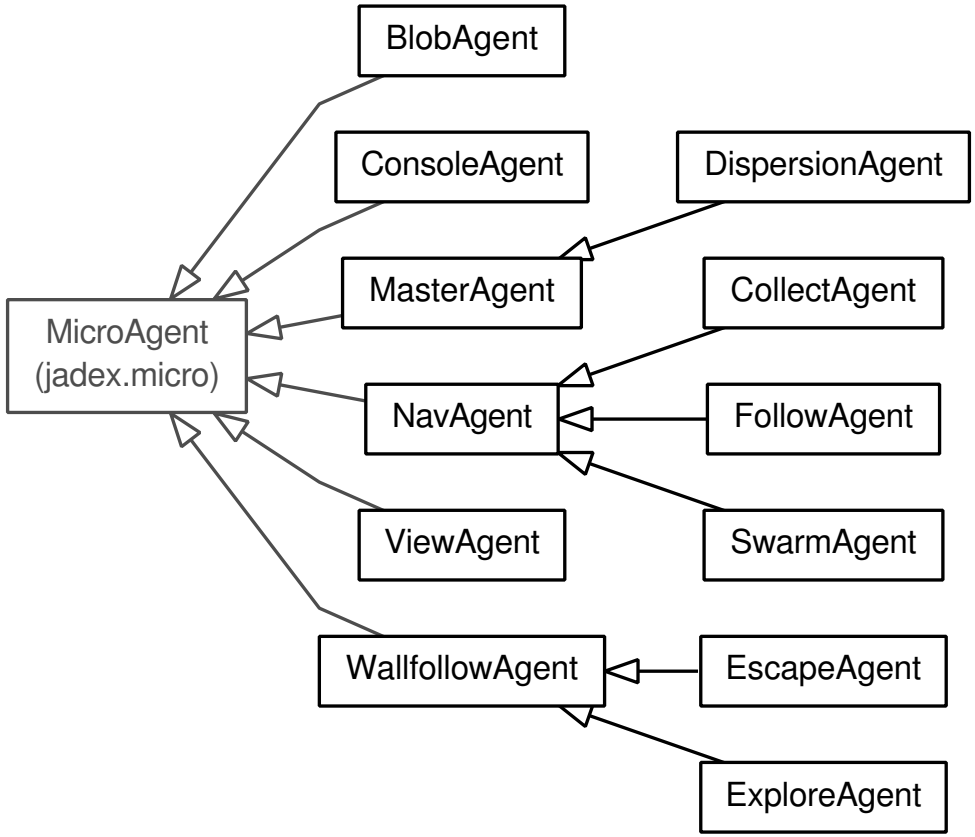


Figure: Example Agents Class Diagram

Hunter and Prey (cont.)

Robot Class Hierarchy

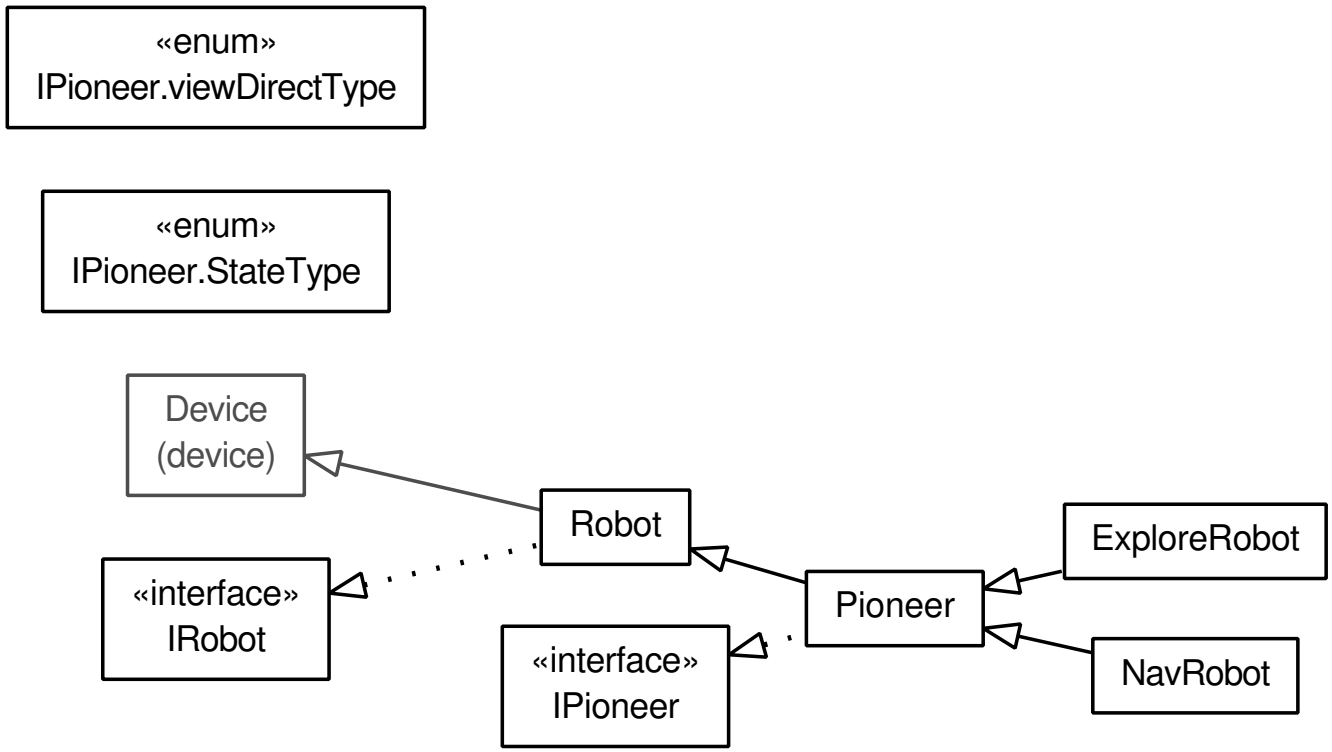


Figure: Example Robots Specialization, »is-a« relation

Hunter and Prey (cont.)

Service Class Hierarchy

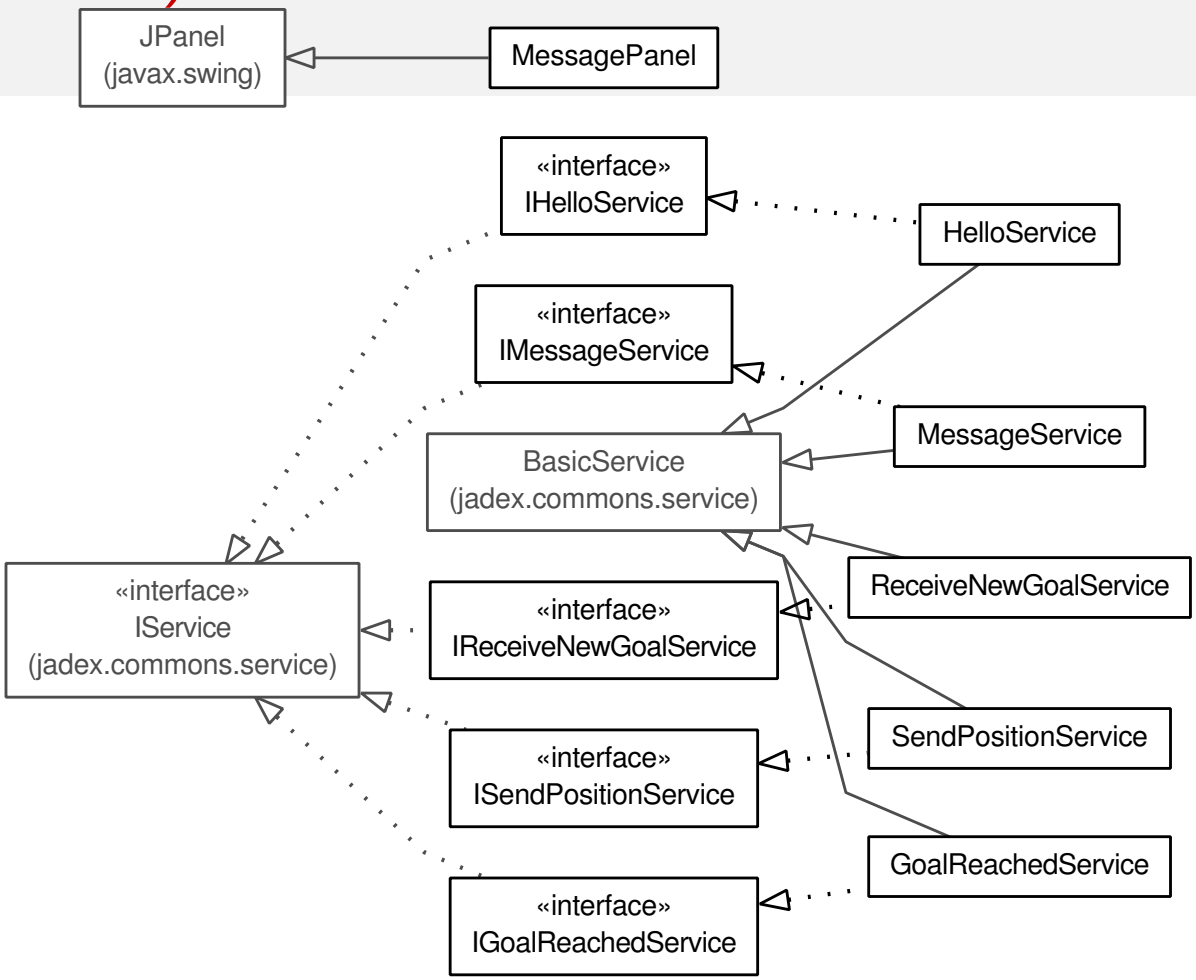


Figure: Example Service Specialization, »is-a« relation

Hunter and Prey (cont.)

Behavior Class Hierarchy

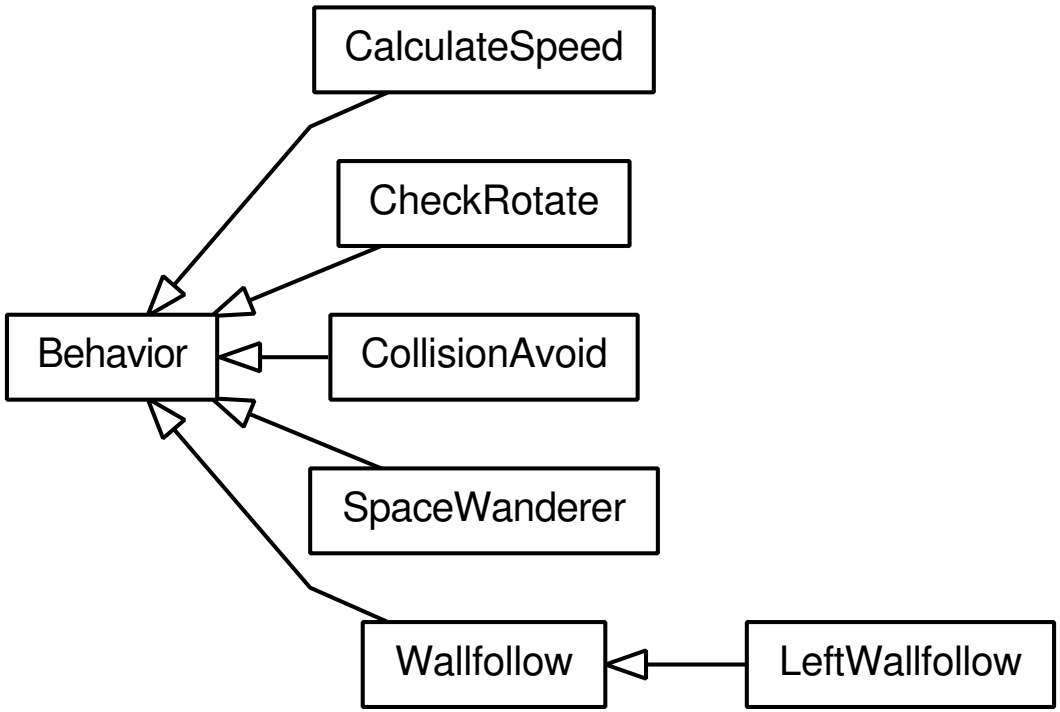


Figure: Example Behavior of Pioneer Robot, »is-a« relation

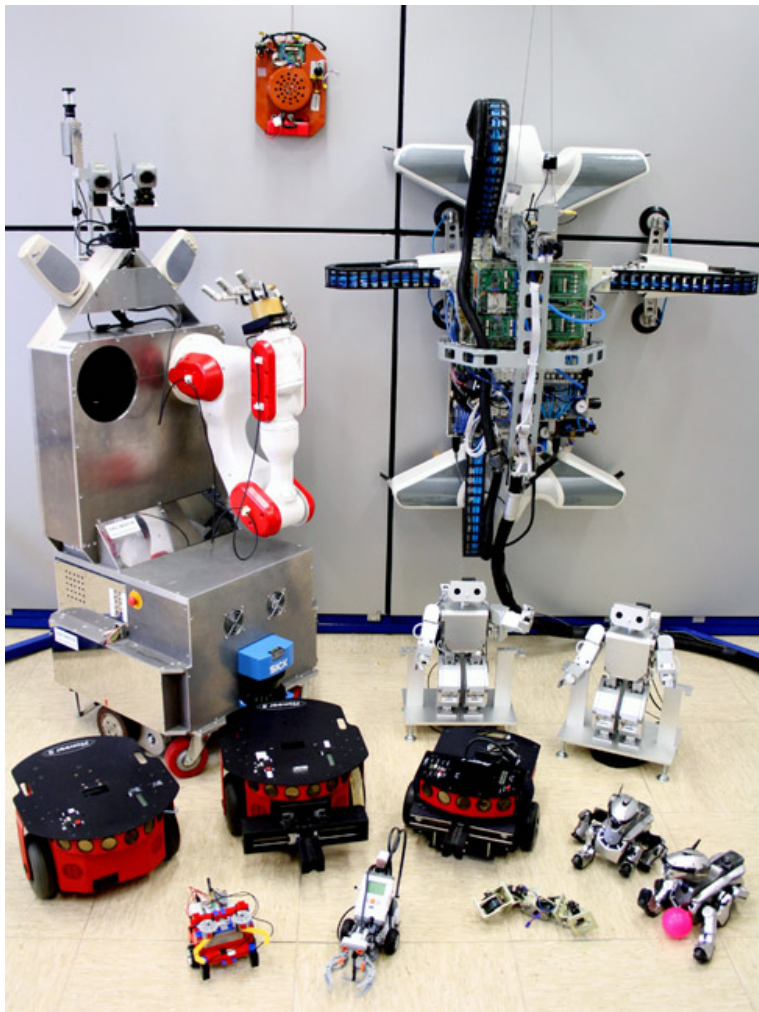
Outline

Outlook

- Thesis
  - Mixed Reality
  - distribution (single vs. multi-hosts)
  - other scenarios
  - tutorial how to write own scenarios
  - evaluation (with Player/Stage)
- Future Projects
  - other planners, behaviors, robots
  - Taser integration
  - cooperative manipulation: Box-pushing
  - cooperative localization/SLAM

Outlook

- Thesis
  - Mixed Reality
  - distribution (single vs. multi-hosts)
  - other scenarios
  - tutorial how to write own scenarios
  - evaluation (with Player/Stage)
- Future Projects
  - other planners, behaviors, robots
  - Taser integration
  - cooperative manipulation: Box-pushing
  - cooperative localization/SLAM



Thank You!

Any questions?

Further Reading

[Bek05] George A. Bekey.
Systems of robots: from cooperation to swarm behavior.
IEEE SMC 2005, Hawaii – Computer Science Department, University of Southern California, December 2005.

[BPL04] Lars Braubach, Alexander Pokahr, and Winfried Lamersdorf.
Jadex: A short overview.
In *Main Conference Net.ObjectDays 2004*, pages 195–207, 9 2004.

Further Reading (cont.)

[Cer08] E. Cervera.
Practical multi-robot applications with player and jade.
In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 2004 –2009, 2008.

[CMW09] I.Y.-H. Chen, B. MacDonald, and B. Wunsche.
Mixed reality simulation for mobile robots.
In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 232 –237, May 2009.

Further Reading (cont.)

[GVH03] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard.
The player/stage project: Tools for multi-robot and distributed sensor systems.
In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal, June 2003.

Further Reading (cont.)

[KNT+09] Ryo Kurazume, Yusuke Noda, Yukihiro Tobata, Kai Lingemann, Yumi Iwashita, and Tsutomu Hasegawa.
Laser-based geometric modeling using cooperative multiple mobile robots.
In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3200 –3205, May 2009.

[QCG+09] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng.
ROS: an open-source robot operating system.
In *ICRA Workshop on Open Source Software*, 2009.

Further Reading (cont.)

[RG10] Sebastian Rockel and Chen Gang.
Mobile robotics project at hamburg university, 2010.

[RN03] S J Russell and Peter Norvig.
Artificial Intelligence, A Modern Approach - 2nd Edition.pdf, volume 82.
Prentice Hall, 2003.

Further Reading (cont.)

[SC09] P.M. Shiroma and M.F.M. Campos.
Comutar: A framework for multi-robot coordination and task allocation.
In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4817 –4824, 2009.

[SMKR09] Xueqing Sun, Tao Mao, J.D. Kralik, and L.E. Ray.
Cooperative multi-robot reinforcement learning: A framework in hybrid state space.
In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1190 –1196, oct. 2009.

Further Reading (cont.)

[SN04] Roland Siegwart and Illah R. Nourbakhsh.
Introduction to Autonomous Mobile Robots.
Bradford Book, 2004.

[SN10] S. Srivastava and G.C. Nandi.
Localization of mobile robots in a network using mobile agents.
In *Computer and Communication Technology (ICCCT), 2010 International Conference on*, pages 415 –420, 2010.

Further Reading (cont.)

[STSI09] Kazuya Suzuki, Tsunamichi Tsukidate, Masahiro Shimizu, and Akio Ishiguro.
Stable and spontaneous self-assembly of a multi-robotic system by exploiting physical interaction between agents.
In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, IROS'09, pages 4343–4348, Piscataway, NJ, USA, 2009. IEEE Press.

Further Reading (cont.)

[TvS10] D. Thomas and O. von Stryk.
Efficient communication in autonomous robot software.
In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 18 - 22 2010.

[RN03][SN04][BPL04][GVH03][QCG+09]