The Fawkes Robot Software Framework

Software Stack for the RoboCup Logistics League

Matthias Löbach
Tim Niemueller

RWTH AACHEN UNIVERSITY

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Carologistics

- RoboCup participant since 2012
- Joint team of FH Aachen University of Applied Sciences & RWTH Aachen University
- Winner of the RoboCup Logistics League 2014, 2015 and 2016

RoboCup Logistics League - At a Glance

RoboCup Logistics League (RCLL)

- In-factory manufacturing logistics in Smart Factory
- Maintain and optimize material flow in production
- Multi-Robot planning/scheduling and coordination

Goals

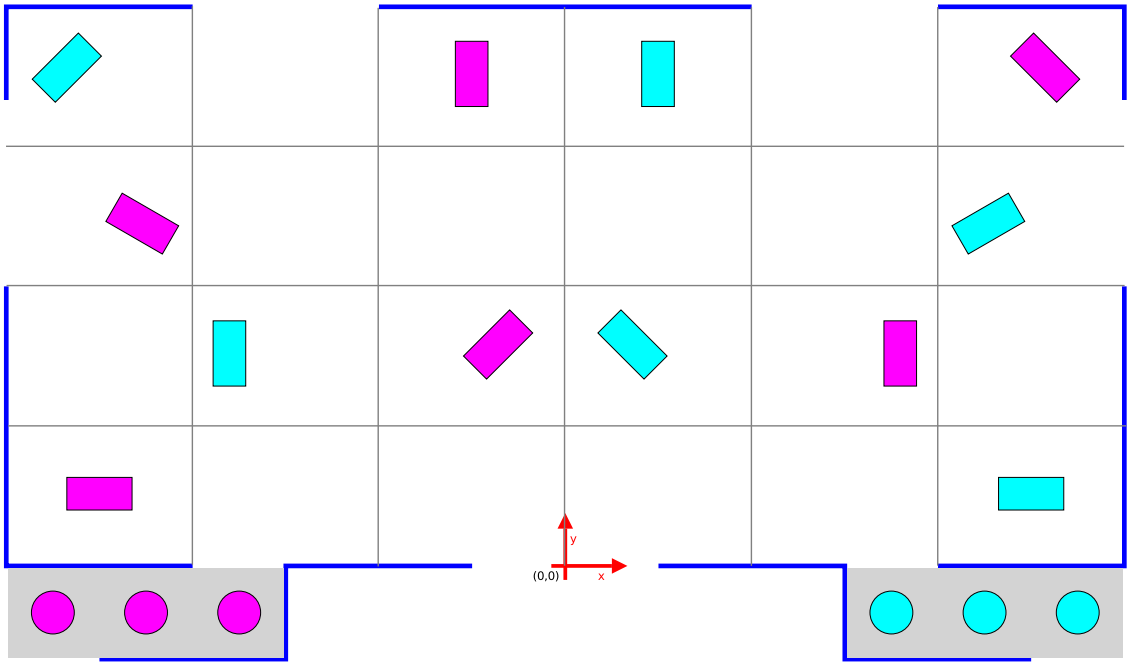
- Production logistics autonomy
- Benchmarking of robots in a Smart Factory
- *Long-term autonomy* with reasoning/planning systems

RoboCup Logistics League

- Physical processing machines based on Festo MPS
- Almost 250 product variants

RoboCup Logistics League

Playing Field



- Team colors: cyan and magenta
- Exclusive machines randomly spread across field
- Mirrored at middle axis

RoboCup Logistics League - Game Phases

Setup Phase

- Duration: 5 minutes
- Teams prepare robots in insertion zone

Exploration Phase

- Duration: 4 minutes
- Points for correct light signal reports
- Points for correct machine position

Production Phase

- Duration: 15 minutes
- Points for delivered products
- Points for intermediate steps

RoboCup Logistics League - Production Machines

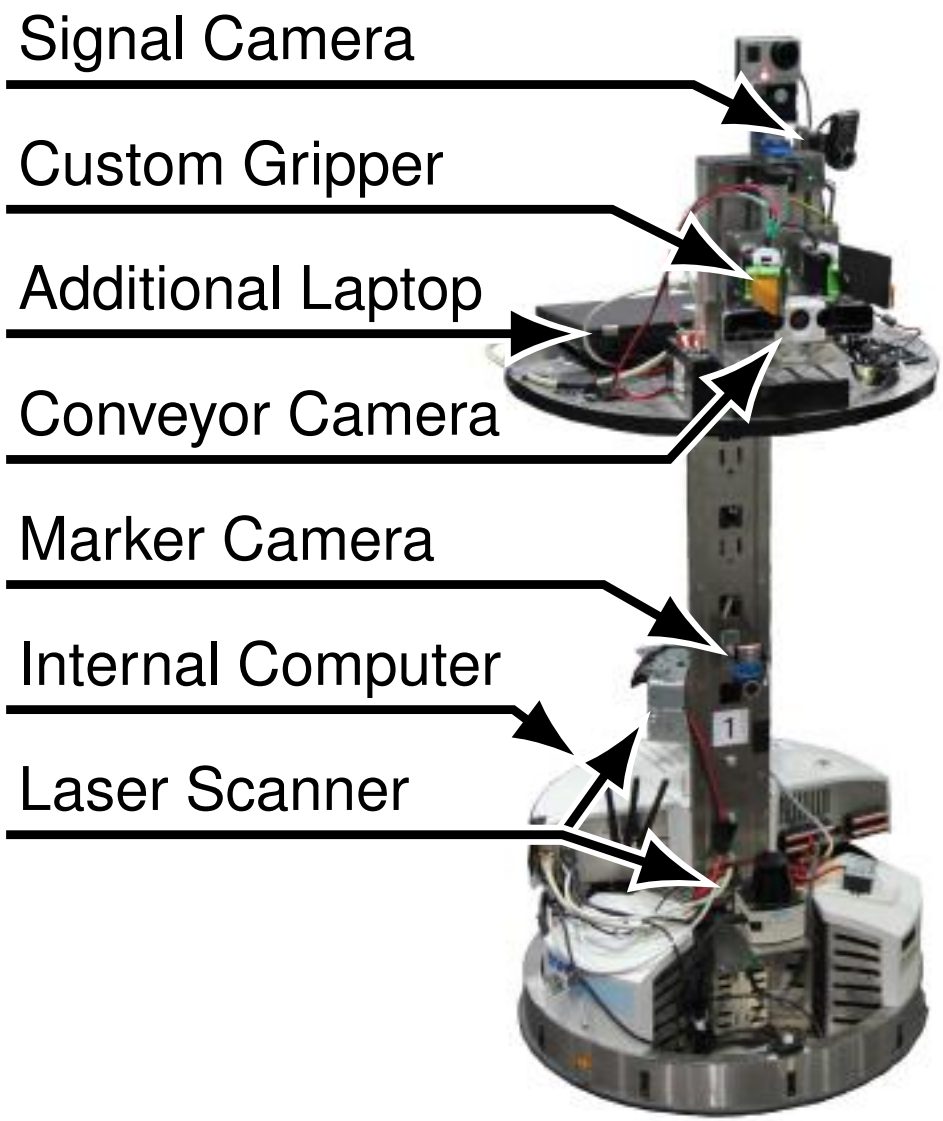
Mobile production machines

- Conveyor as contact point for robots
- AR-Tag for identification and localization
- Signal light for identification and status

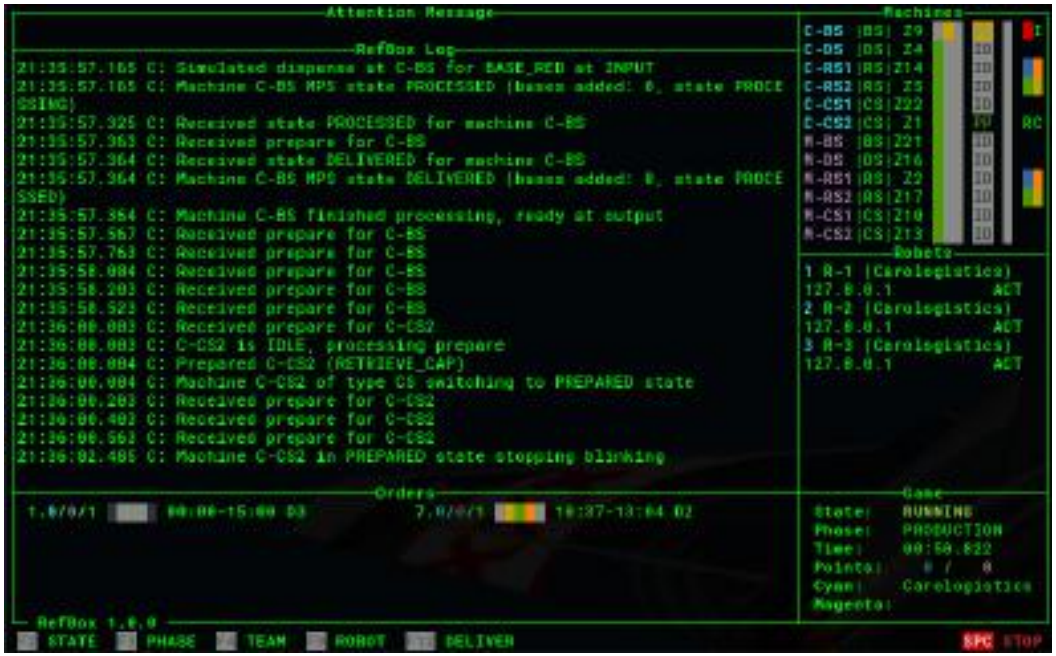
RoboCup Logistics League - Robots

Robotino 3 platform

- Laser range finder
- Additional laptop
- Custom gripper
- Cameras



RoboCup Logistics League - Referee Box



Game Control

- Maintain game state/score

Communication

- Publish production plans

Data Recording

- Collect benchmarking data

Visualization and Instruction

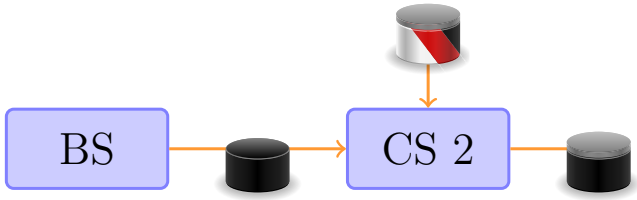
- Referee/visitor monitoring

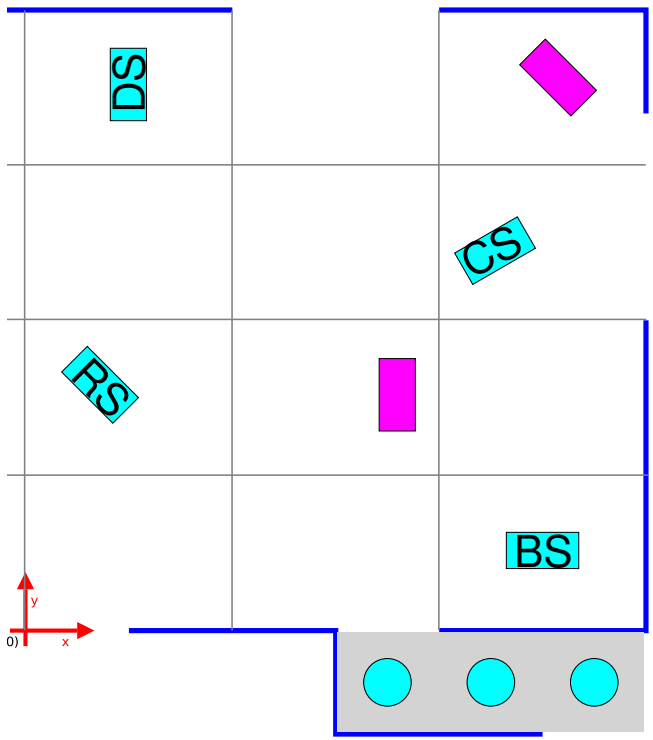
Machinery Control

- Instruct field machines

Logs game information and all communication

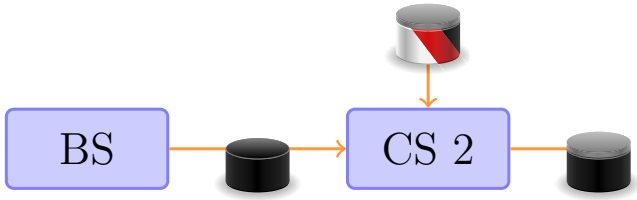
RoboCup Logistics League – Production Example

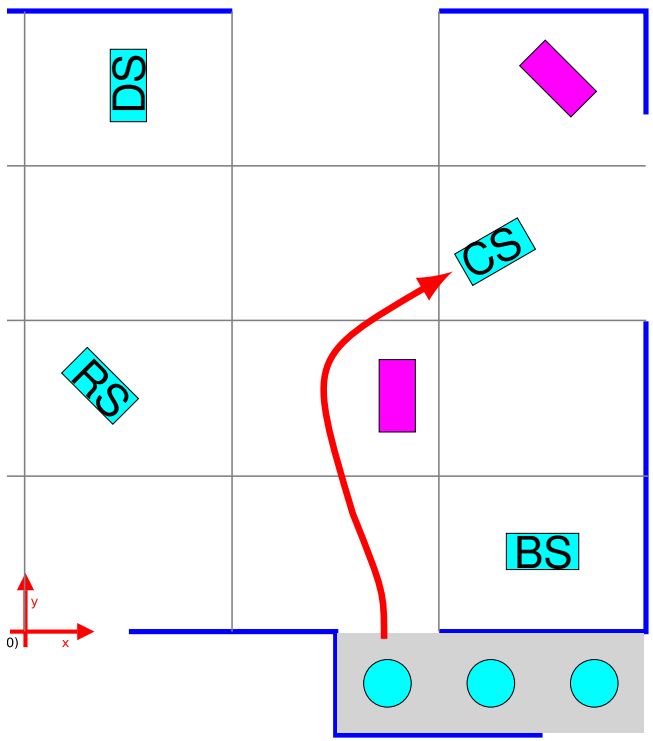




RoboCup Logistics League – Production Example

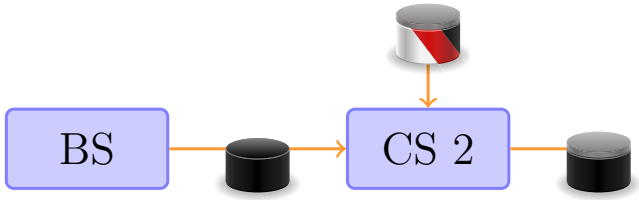
- Get base with cap

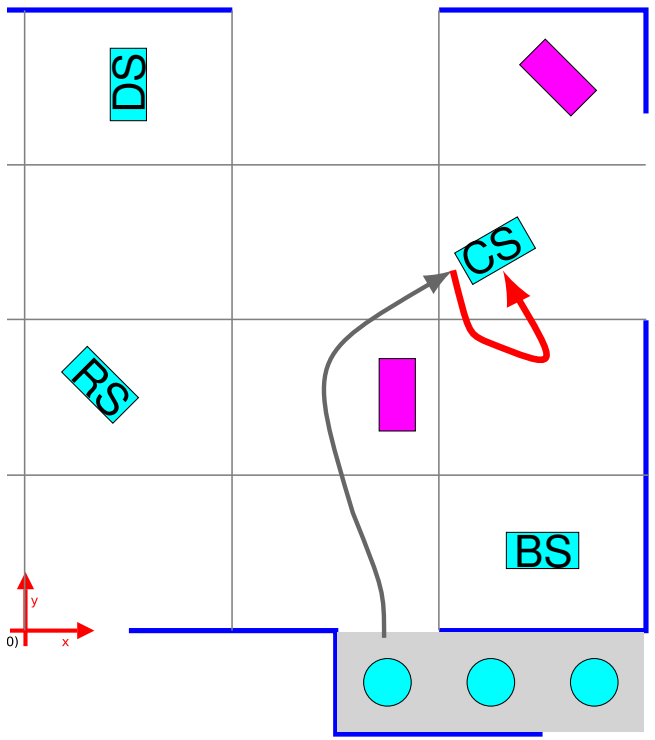




RoboCup Logistics League – Production Example

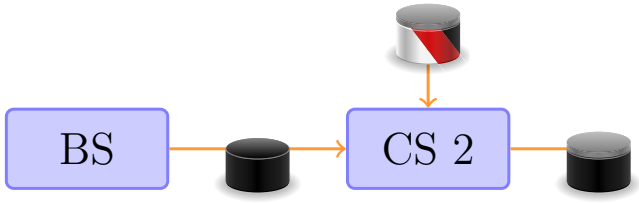
- Get base with cap
- Fill cap station with cap

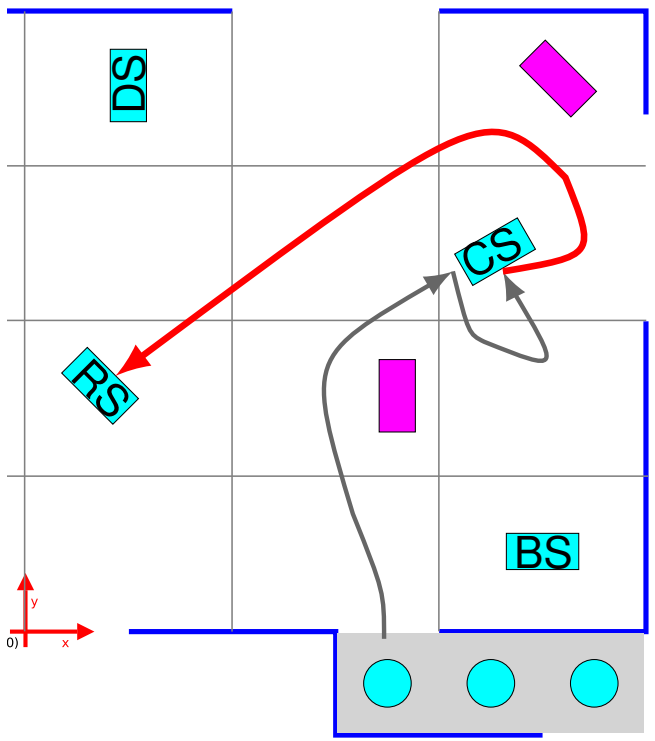




RoboCup Logistics League – Production Example

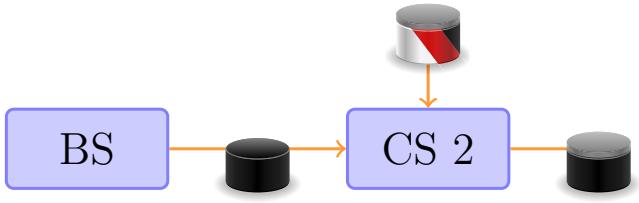
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource

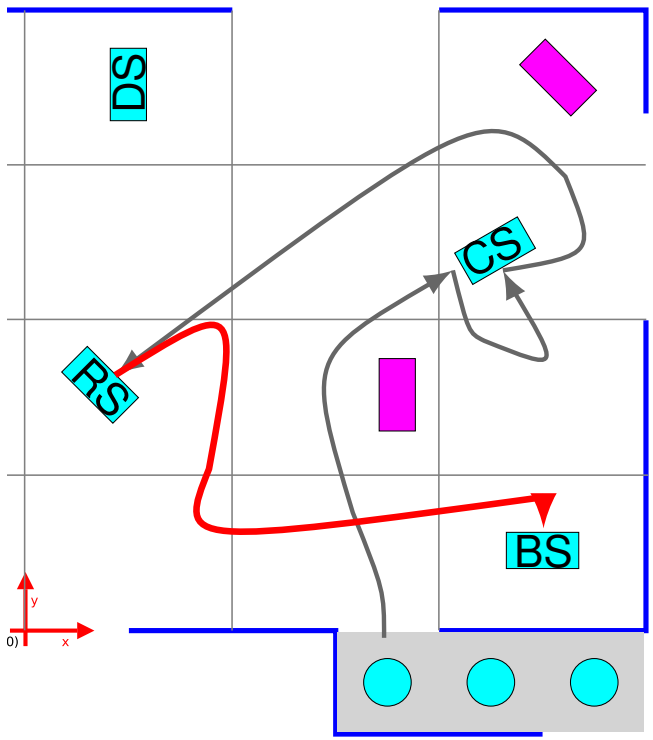




RoboCup Logistics League – Production Example

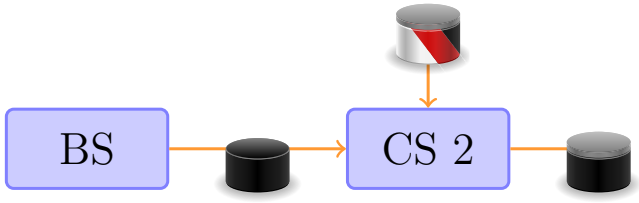
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station

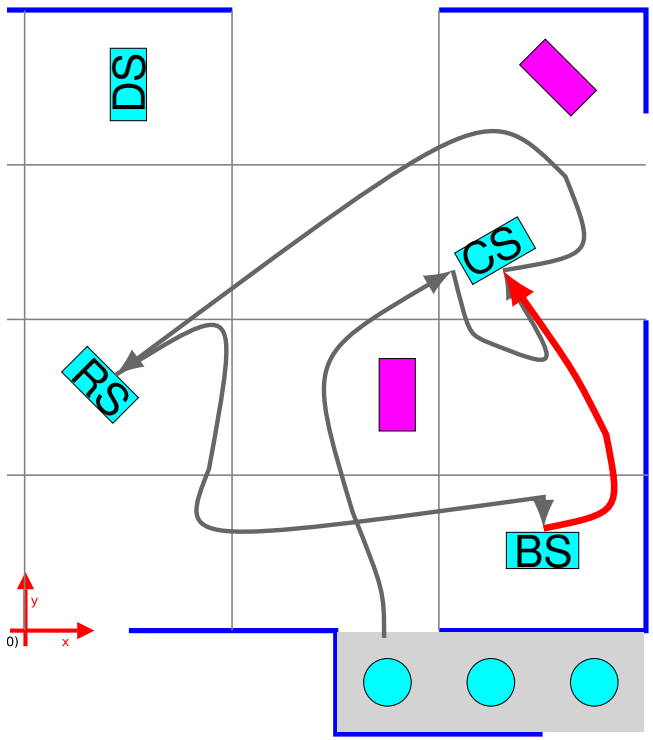




RoboCup Logistics League – Production Example

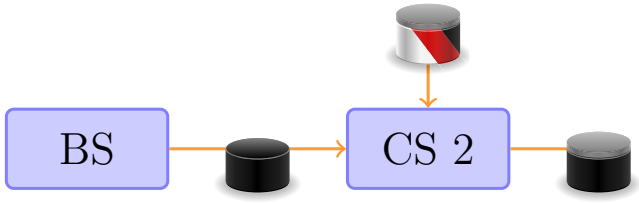
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station
- Insert base into cap station for assembly

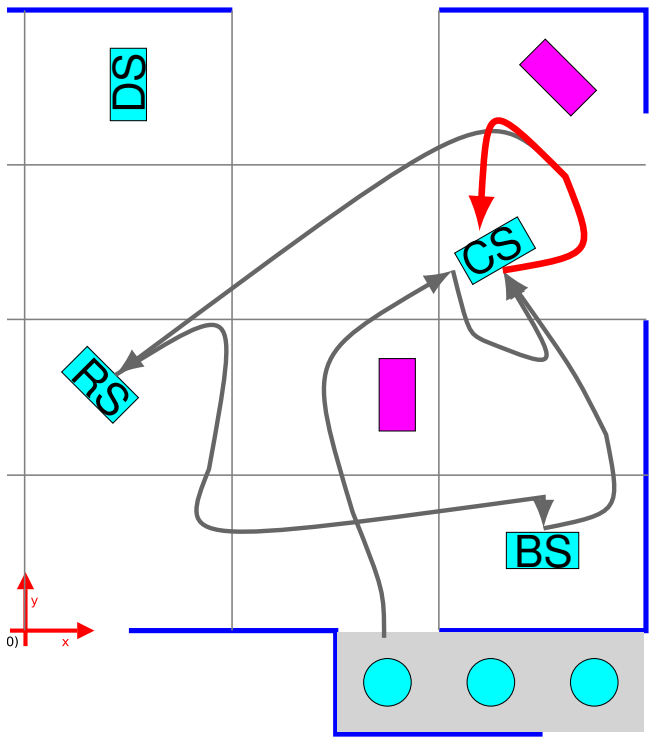




RoboCup Logistics League – Production Example

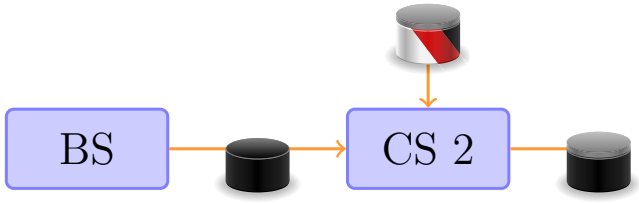
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station
- Insert base into cap station for assembly
- Retrieve assembled product from cap station

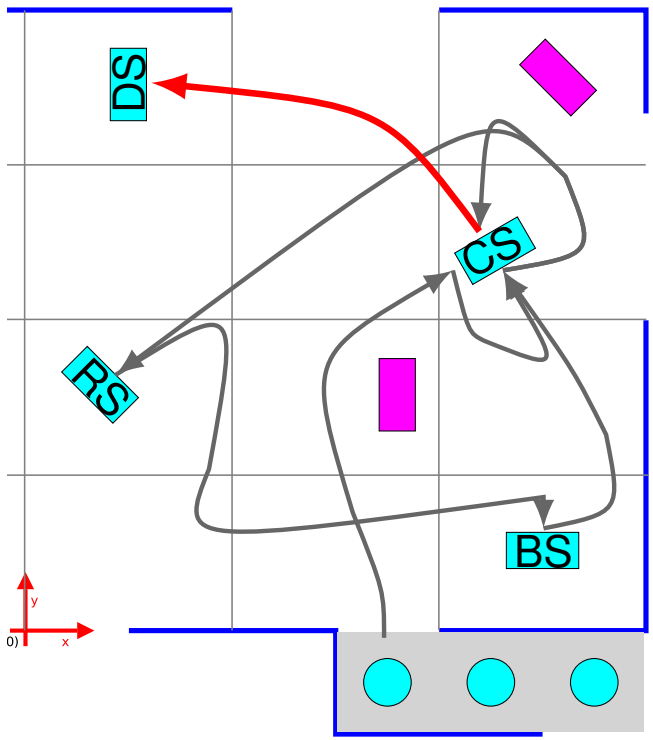




RoboCup Logistics League – Production Example

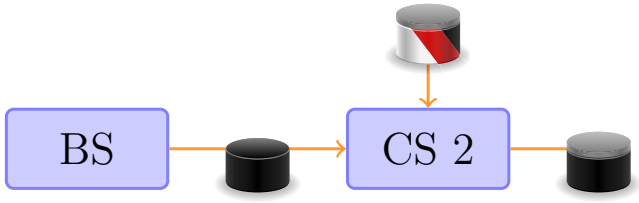
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station
- Insert base into cap station for assembly
- Retrieve assembled product from cap station
- Deliver product at delivery station

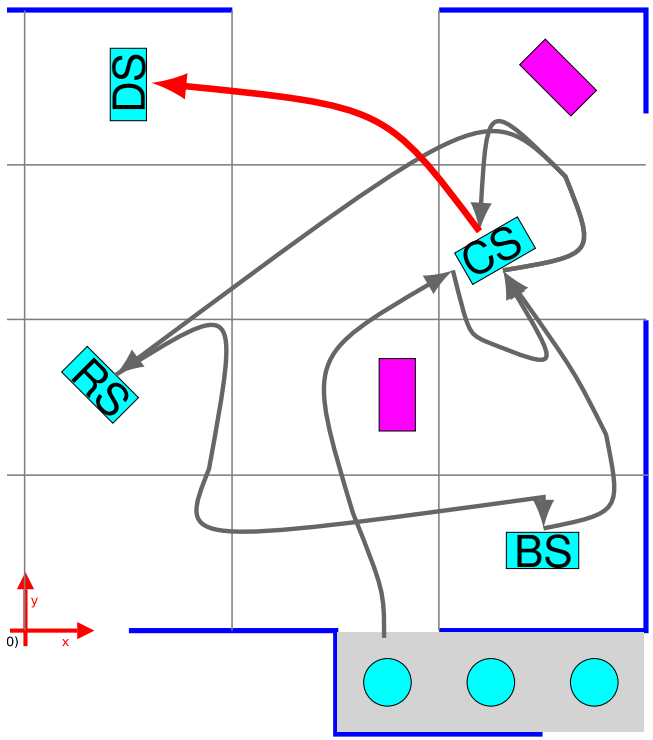




RoboCup Logistics League – Production Example

- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station
- Insert base into cap station for assembly
- Retrieve assembled product from cap station
- Deliver product at delivery station

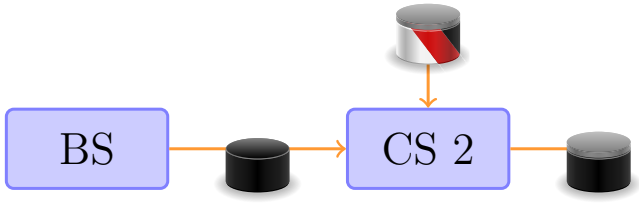


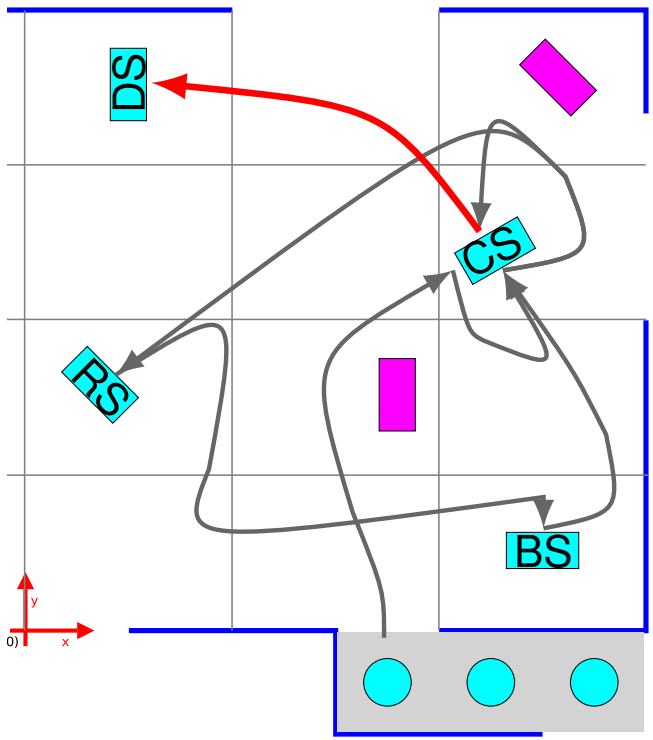


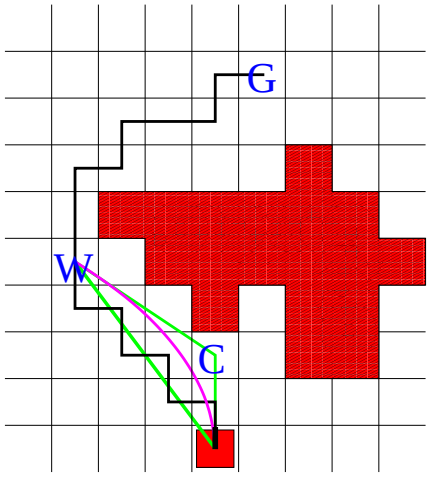
RoboCup Logistics League – Production Example

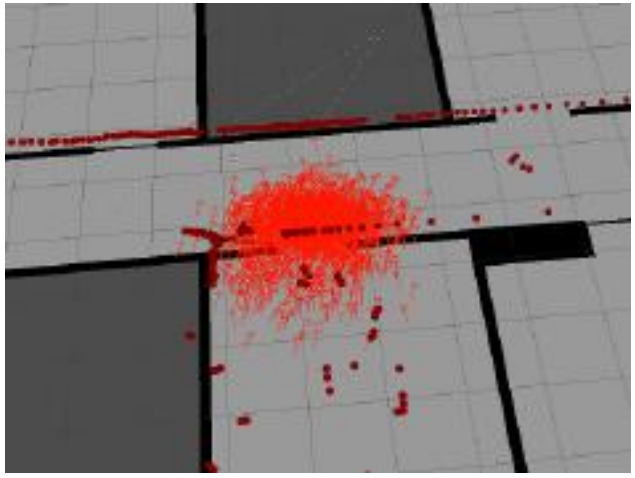
- Get base with cap
- Fill cap station with cap
- Bring surplus base to ring station as resource
- Retrieve base with ordered color at base station
- Insert base into cap station for assembly
- Retrieve assembled product from cap station
- Deliver product at delivery station

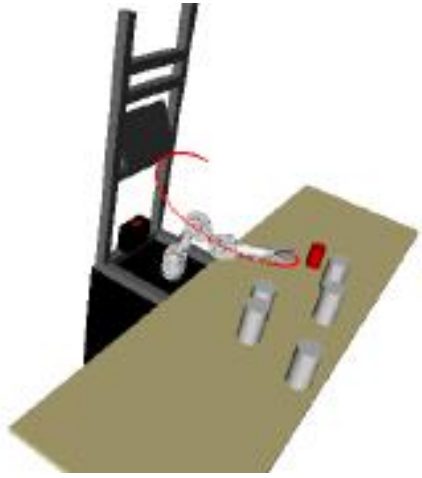
Most simple product has seven steps

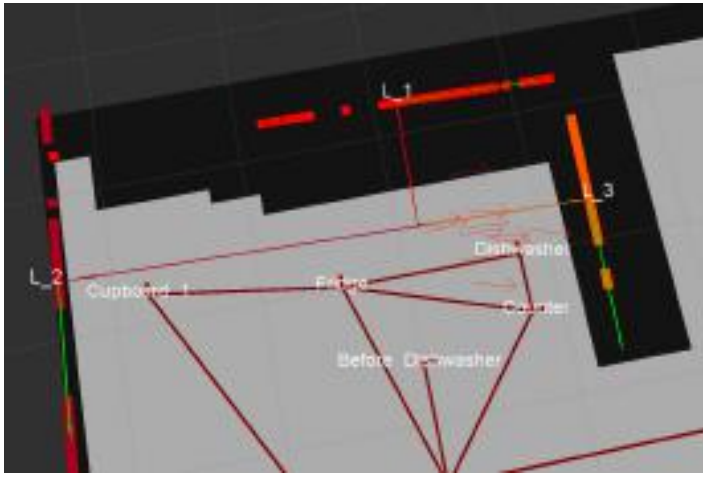


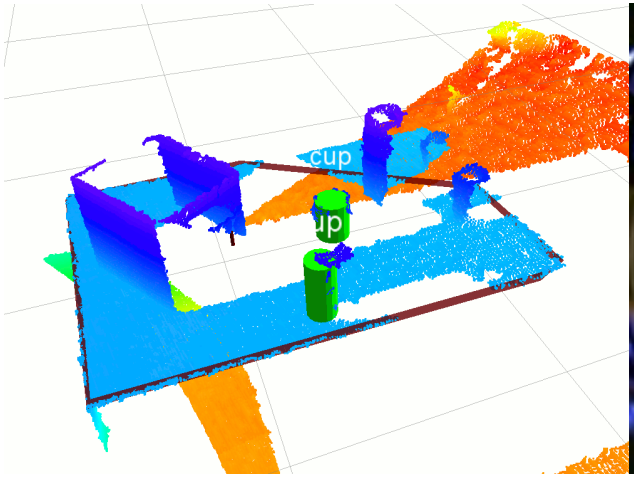


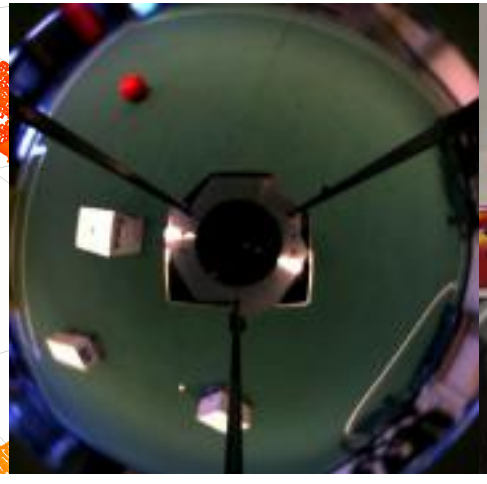

G
W
C

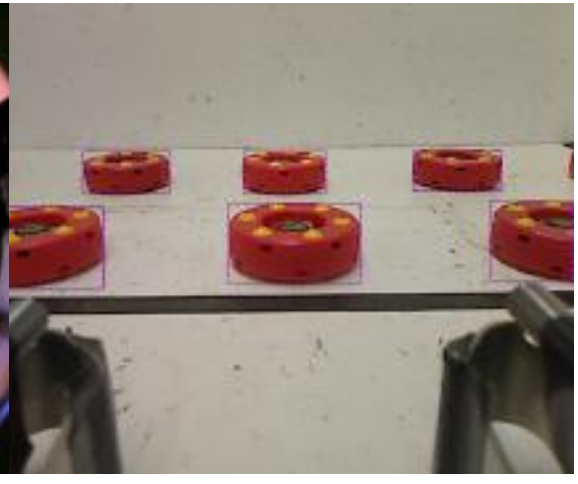





L_1
L_2
L_3
Dishwasher
Before Dishwasher


cup





Integration

Middleware

- Interconnect software components
- Make data accessible and observable
- Structure the data

Integration

Middleware

- Interconnect software components
- Make data accessible and observable
- Structure the data

Framework

- Run-time organization and execution flow
- Assert certain properties of the system

Integration

Middleware

- Interconnect software components
- Make data accessible and observable
- Structure the data

Framework

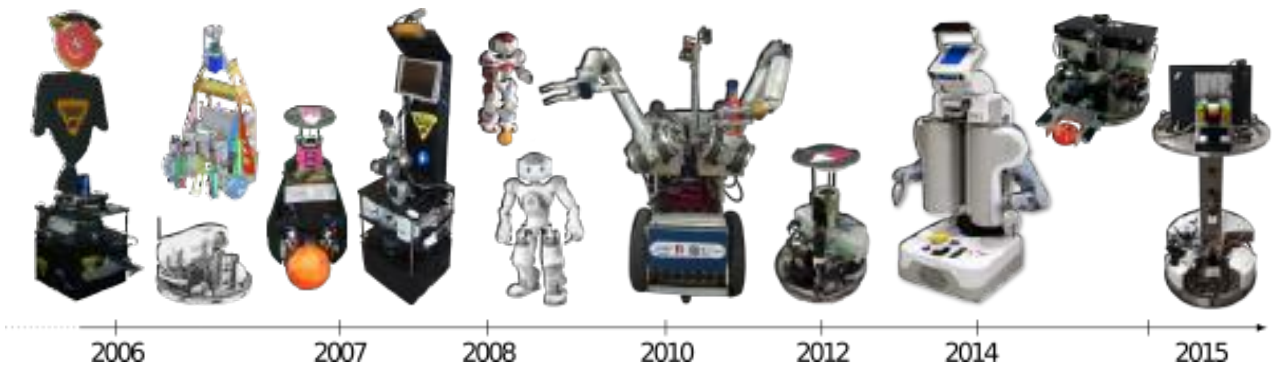
- Run-time organization and execution flow
- Assert certain properties of the system

Toolbox

- Provide libraries for typical robotics task
- Integrate libraries and make available through framework

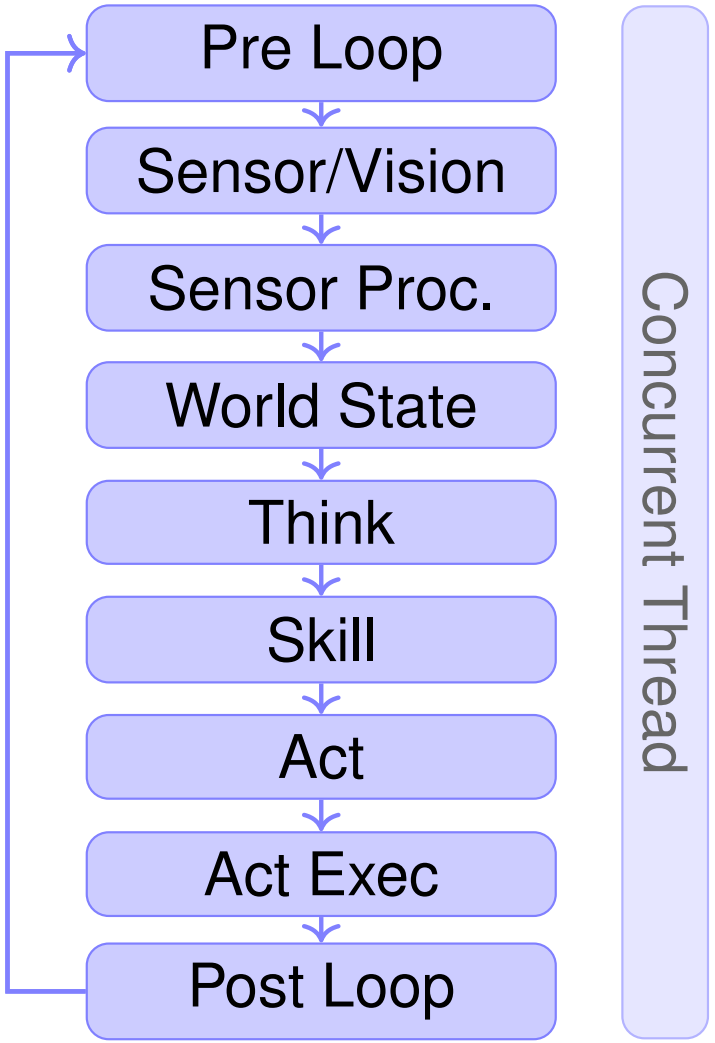
Fawkes

- *Component-based* architecture (plugins)
- Hybrid *BlackBoard/messaging* data exchange
- Multi-threaded and distributable
- Aspect-oriented framework feature access
- Structured and synchronized main loop
- Automated coordinate transform system
- Web interface for introspection and control



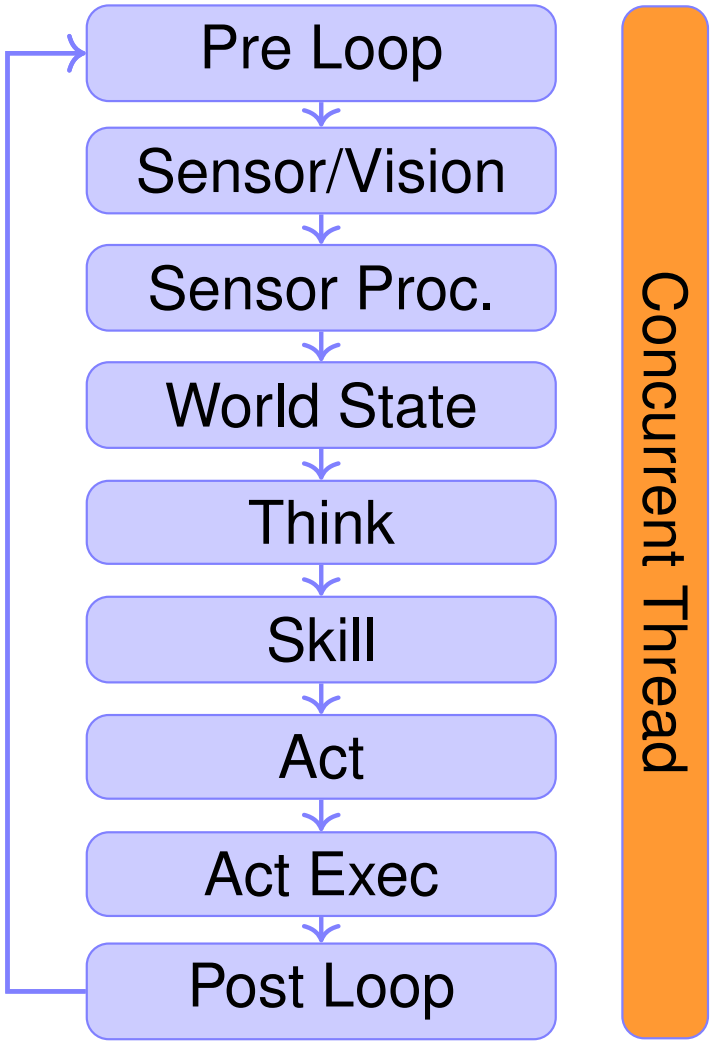
Run-time Coordination

- Fawkes provides a main loop
- Threads *can* be hooked into the main loop
- Threads *can* also run concurrently
- Main loop is replaceable
- Threads for each hook are woken up concurrently
- Threads sleep during execution of other hooks

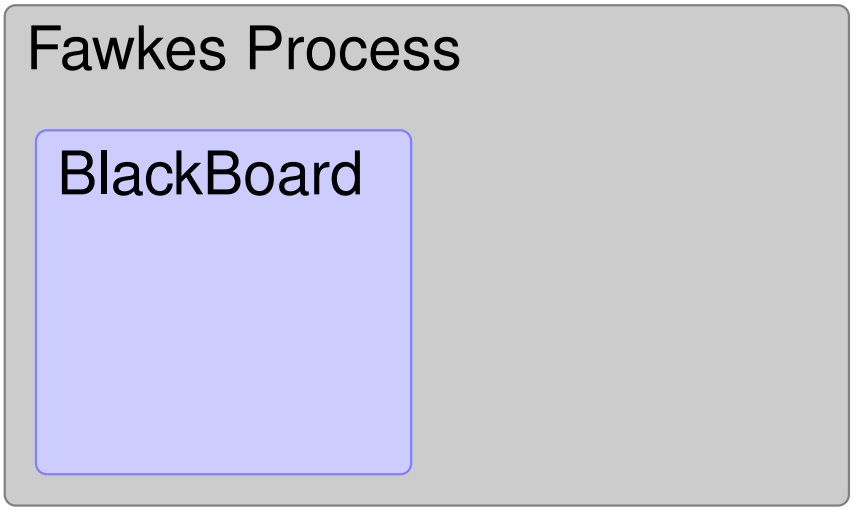


Run-time Coordination

- Fawkes provides a main loop
- Threads *can* be hooked into the main loop
- Threads *can* also run concurrently
- Main loop is replaceable
- Threads for each hook are woken up concurrently
- Threads sleep during execution of other hooks

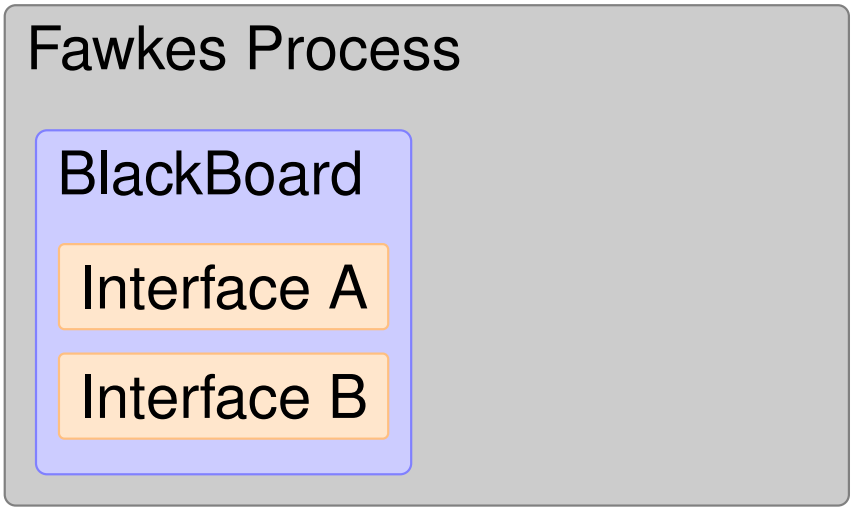


BlackBoard



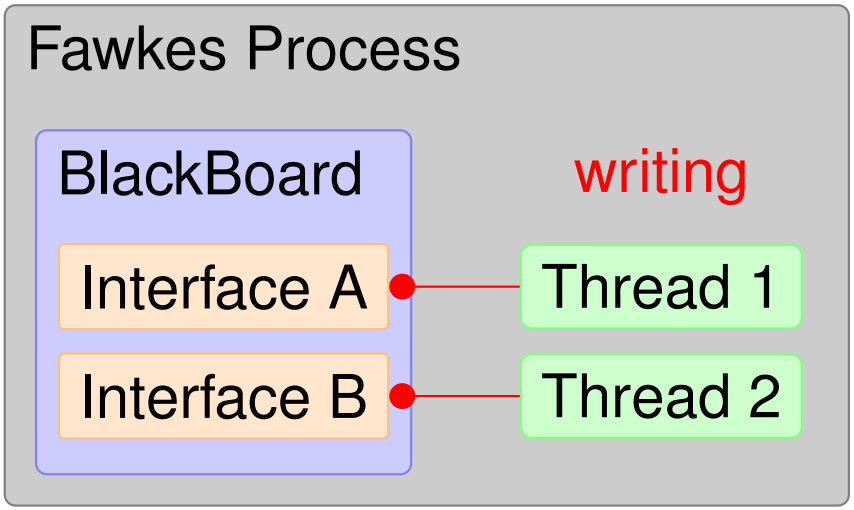
- BlackBoard created by Fawkes main application

BlackBoard



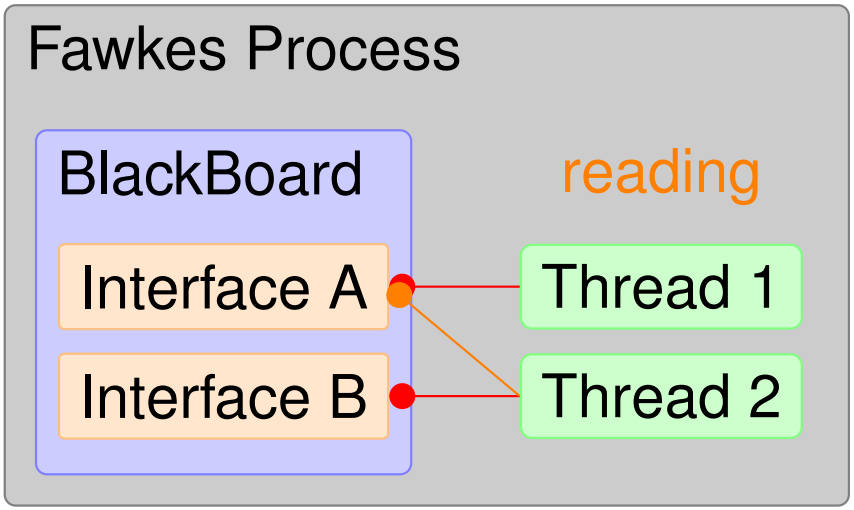
- BlackBoard created by Fawkes main application
- Interface storage in the BlackBoard memory
- Interface definition via XML (fields/messages)

BlackBoard



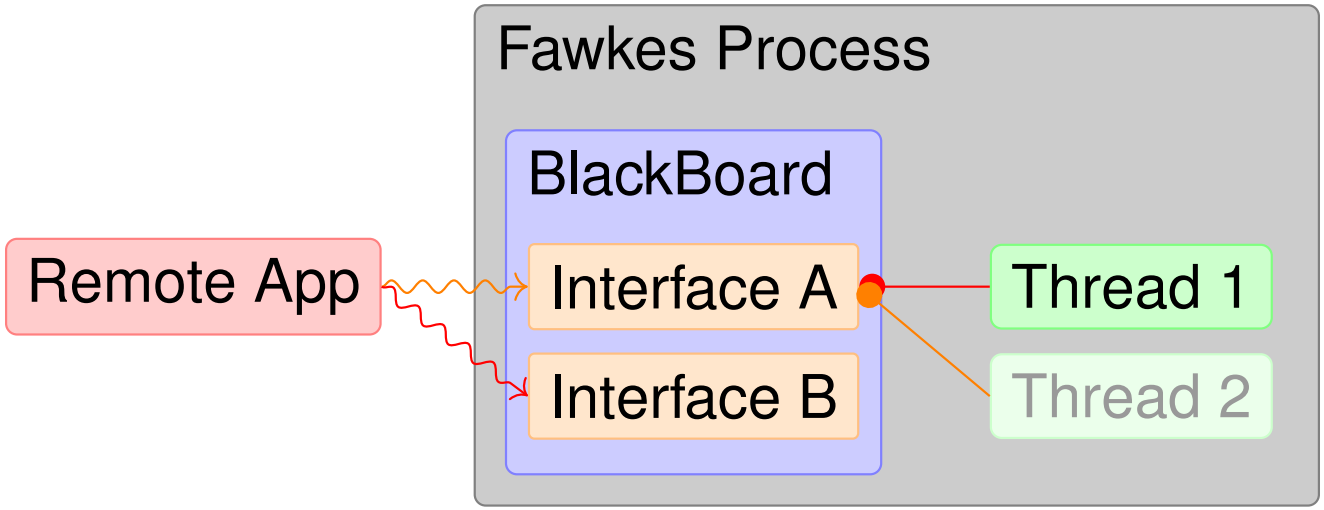
- Fawkes Threads access the BlackBoard via these Interfaces

BlackBoard



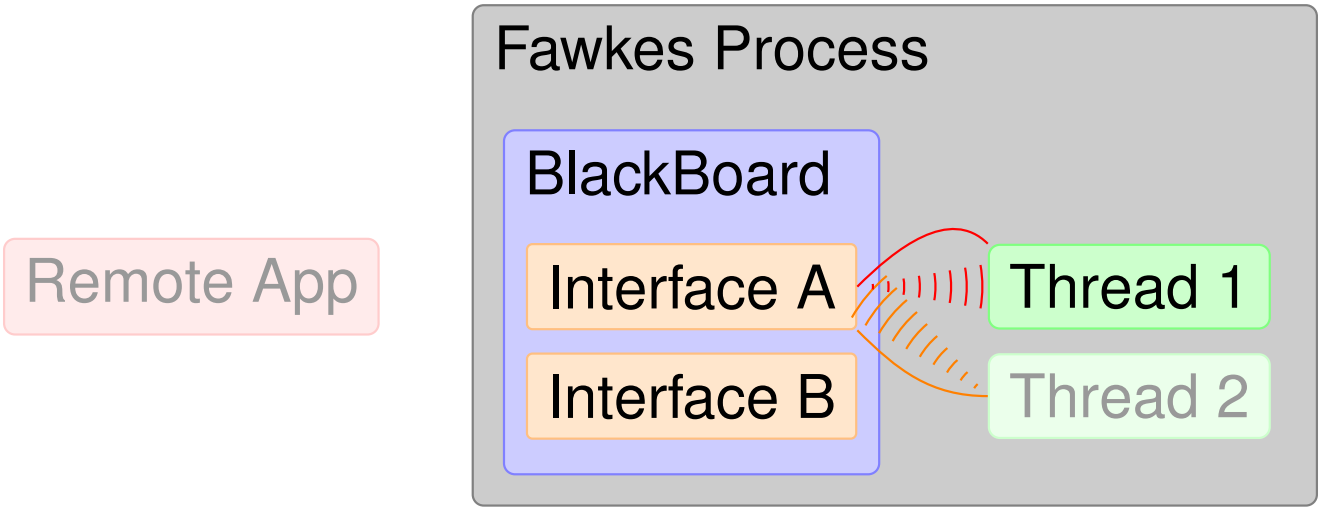
- Fawkes Threads access the BlackBoard via these Interfaces

BlackBoard



- Fawkes Threads access the BlackBoard via these Interfaces
- Remote applications can access BlackBoard via network
- Transactions (read/write)
- Only one writer at a time

BlackBoard

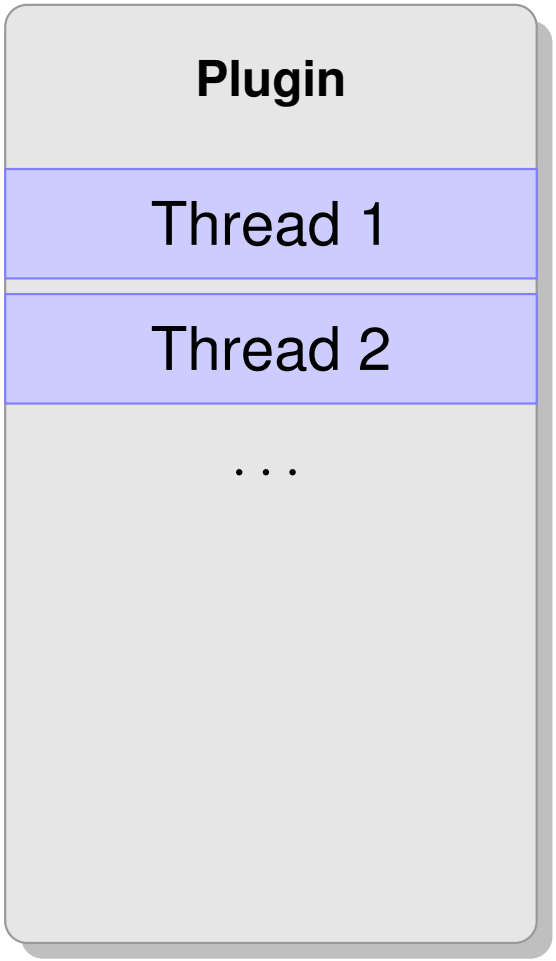


- Message passing as command channel
- Messages can only be sent from reader to writer
- Any number of messages in queue

Plugins

Components

- Provides specific functionality
- Ideally: parameterizable blackbox
- Can – ideally – be easily replaced
- Interconnected through middleware



Plugins

Components

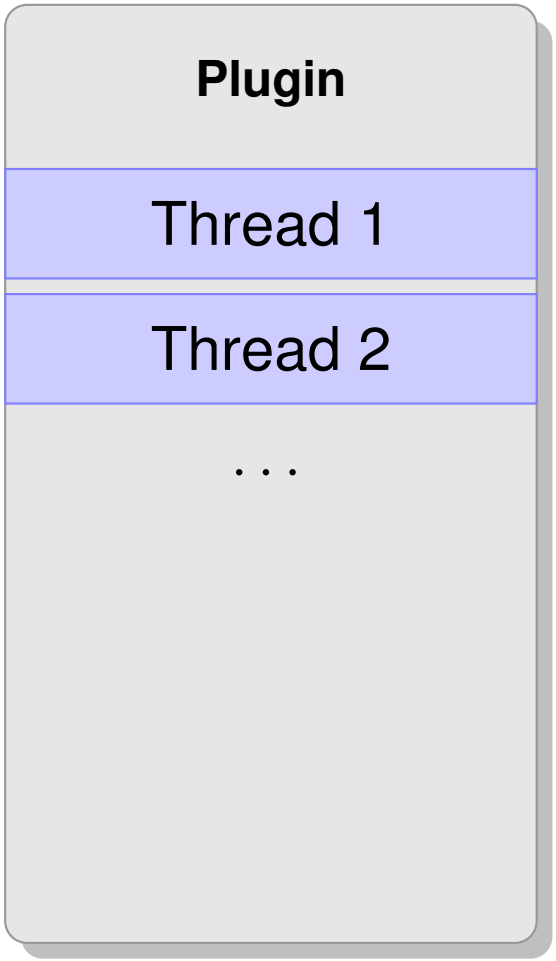
- Provides specific functionality
- Ideally: parameterizable blackbox
- Can – ideally – be easily replaced
- Interconnected through middleware

Plugins

- Dynamically loadable libraries
- Set of threads and their initialization
- Framework can introspect threads

Soft Guarantee for Plugins

⇒ Either all threads are successfully initialized, or none is ever started



Plugins

Components

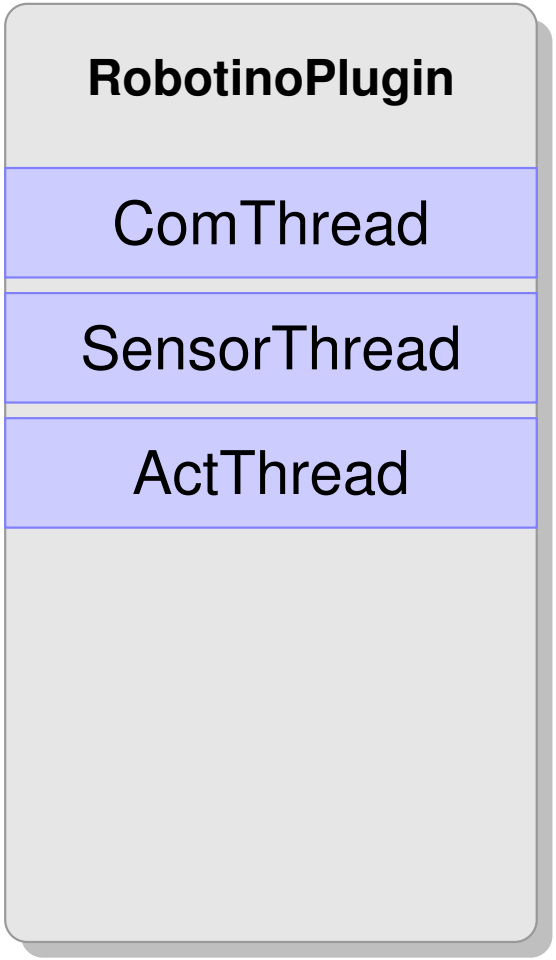
- Provides specific functionality
- Ideally: parameterizable blackbox
- Can – ideally – be easily replaced
- Interconnected through middleware

Plugins

- Dynamically loadable libraries
- Set of threads and their initialization
- Framework can introspect threads

Soft Guarantee for Plugins

⇒ Either all threads are successfully initialized, or none is ever started



Plugins

Components

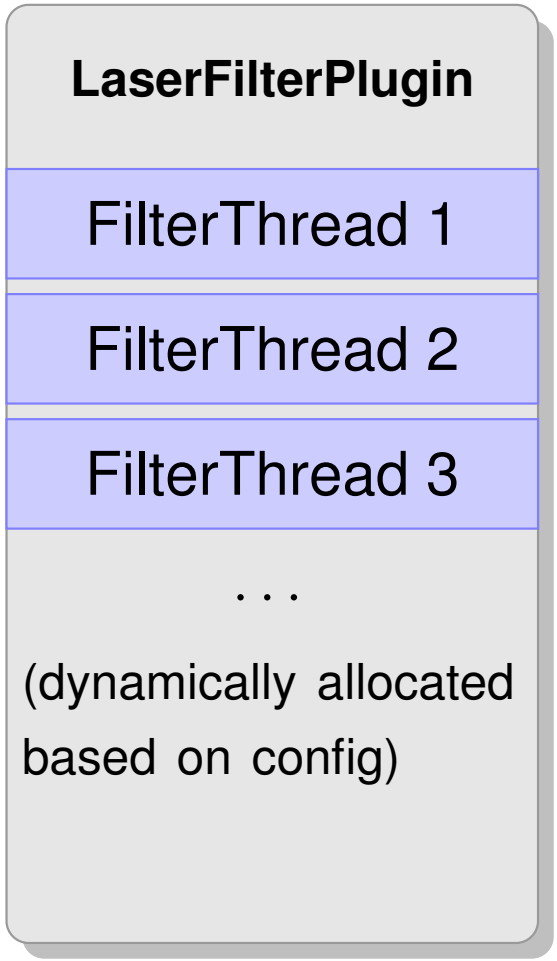
- Provides specific functionality
- Ideally: parameterizable blackbox
- Can – ideally – be easily replaced
- Interconnected through middleware

Plugins

- Dynamically loadable libraries
- Set of threads and their initialization
- Framework can introspect threads

Soft Guarantee for Plugins

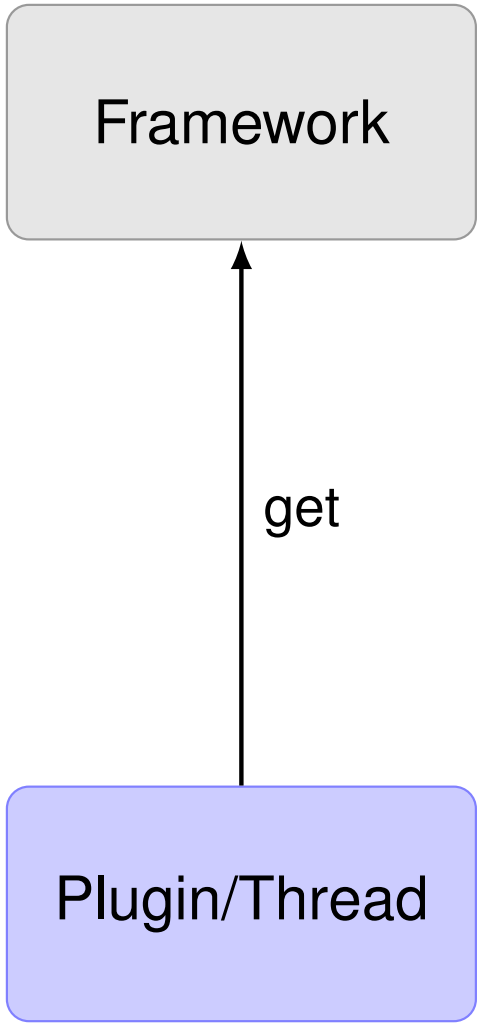
⇒ Either all threads are successfully initialized, or none is ever started



Framework Features

- Threads must access features
- Classic: inquire/get features
- Control executed by requester

⇒ Framework has only limited information



Aspects

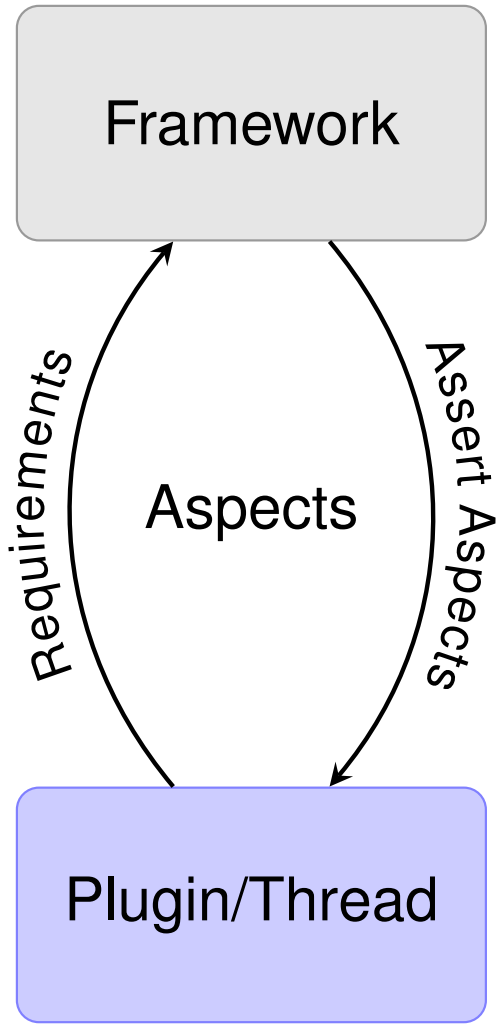
Framework Features

- Threads must access features
- Classic: inquire/get features
- Control executed by requester

$\Rightarrow$ Framework has only limited information

Aspects

- Plugin threads denote requirements
- Framework initializes aspects
- Soft guarantee of feature availability



Aspects

Framework Features

- Threads must access features
- Classic: inquire/get features
- Control executed by requester

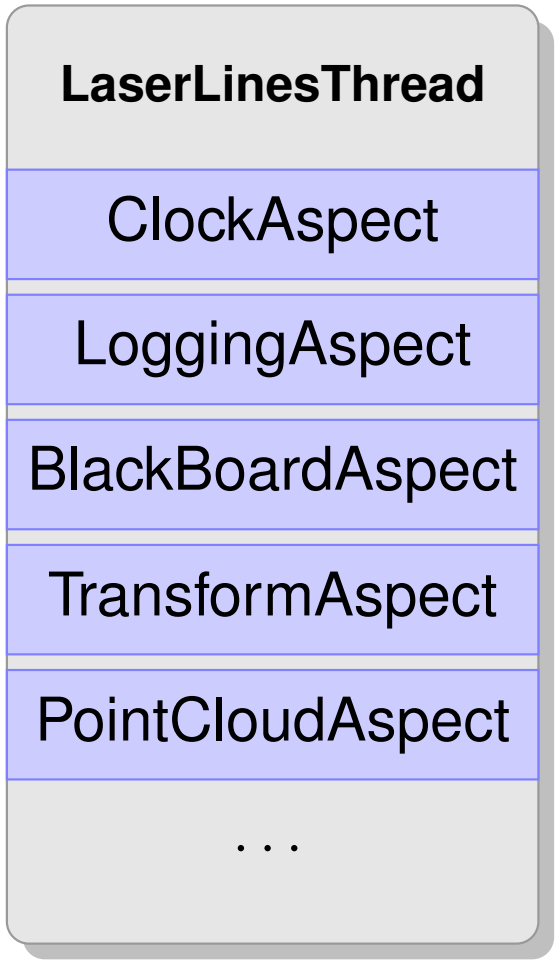
⇒ Framework has only limited information

Aspects

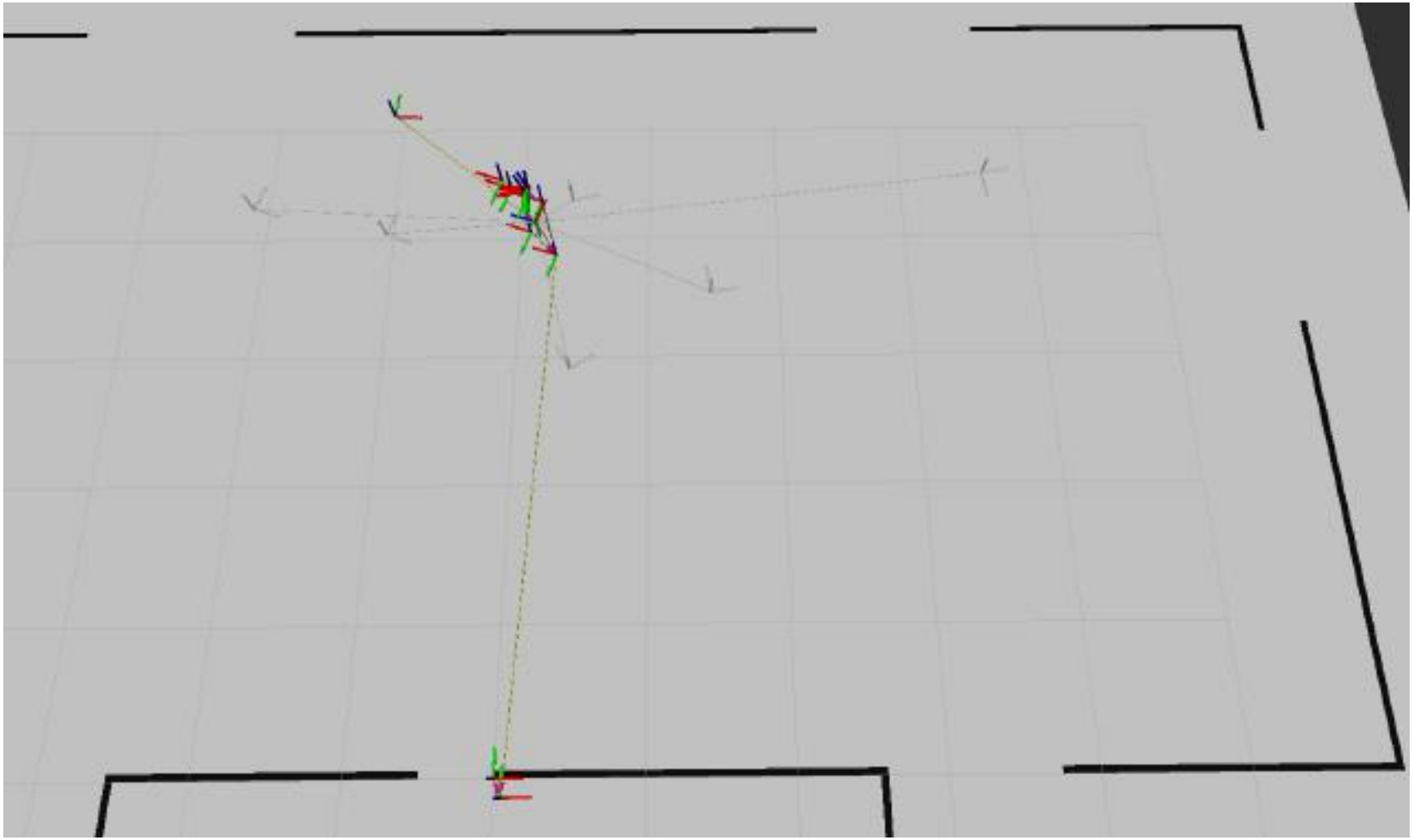
- Plugin threads denote requirements
- Framework initializes aspects
- Soft guarantee of feature availability

Implementation

- Aspect as simple class
- Threads inherit aspect class
- Framework asserts initialization



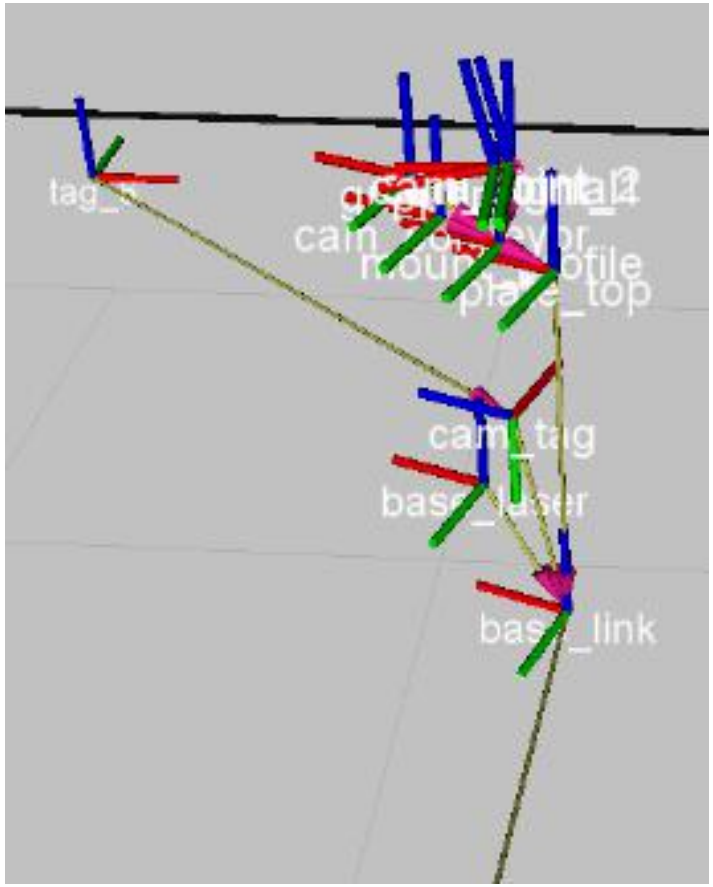
Transforms



Transforms

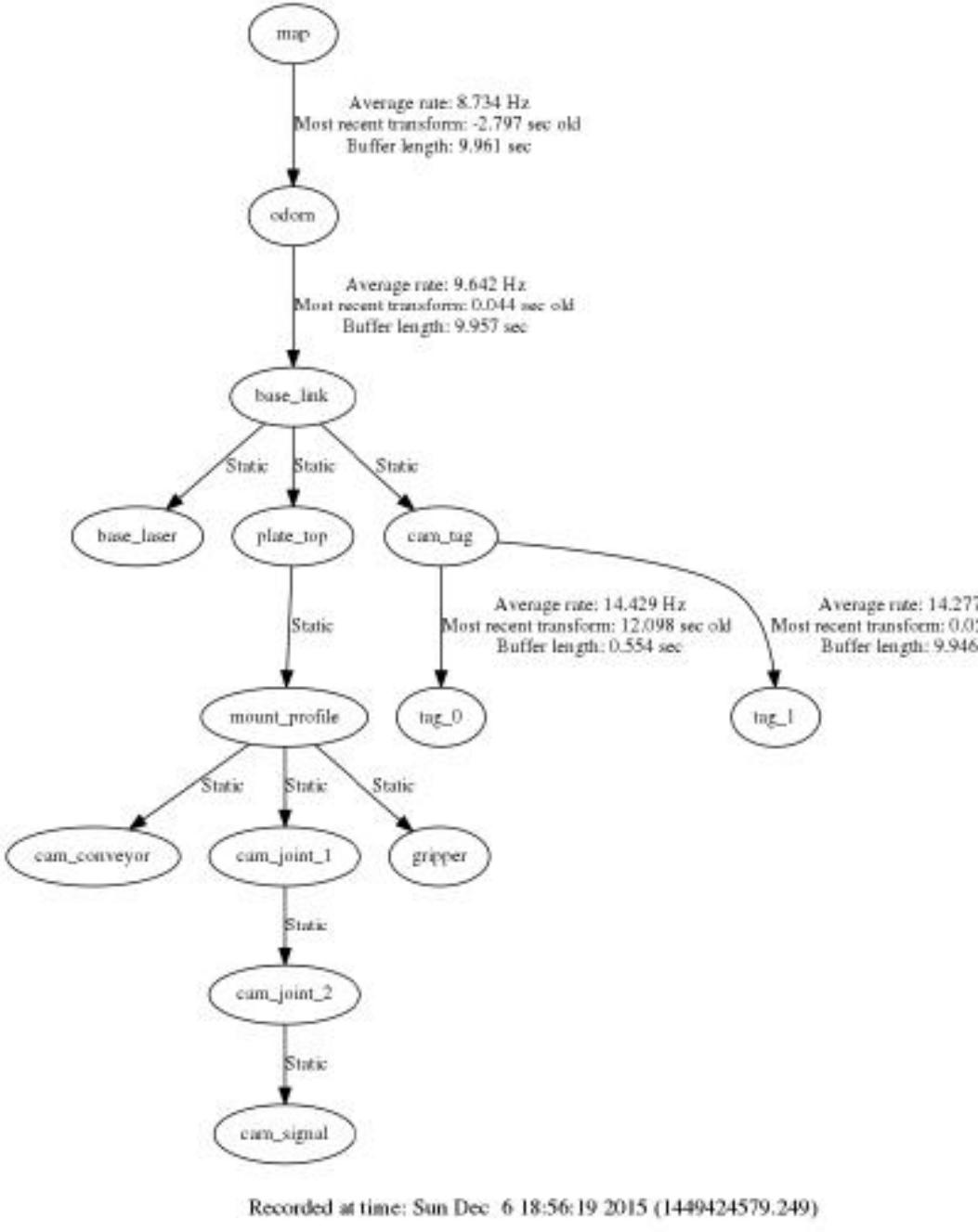
- Various coordinate reference frames
- Keep data in local frame for accuracy
- Need transformations to act on sensed objects

- Generalized transformation system (ported from ROS tf2)
- Tree of linked frames

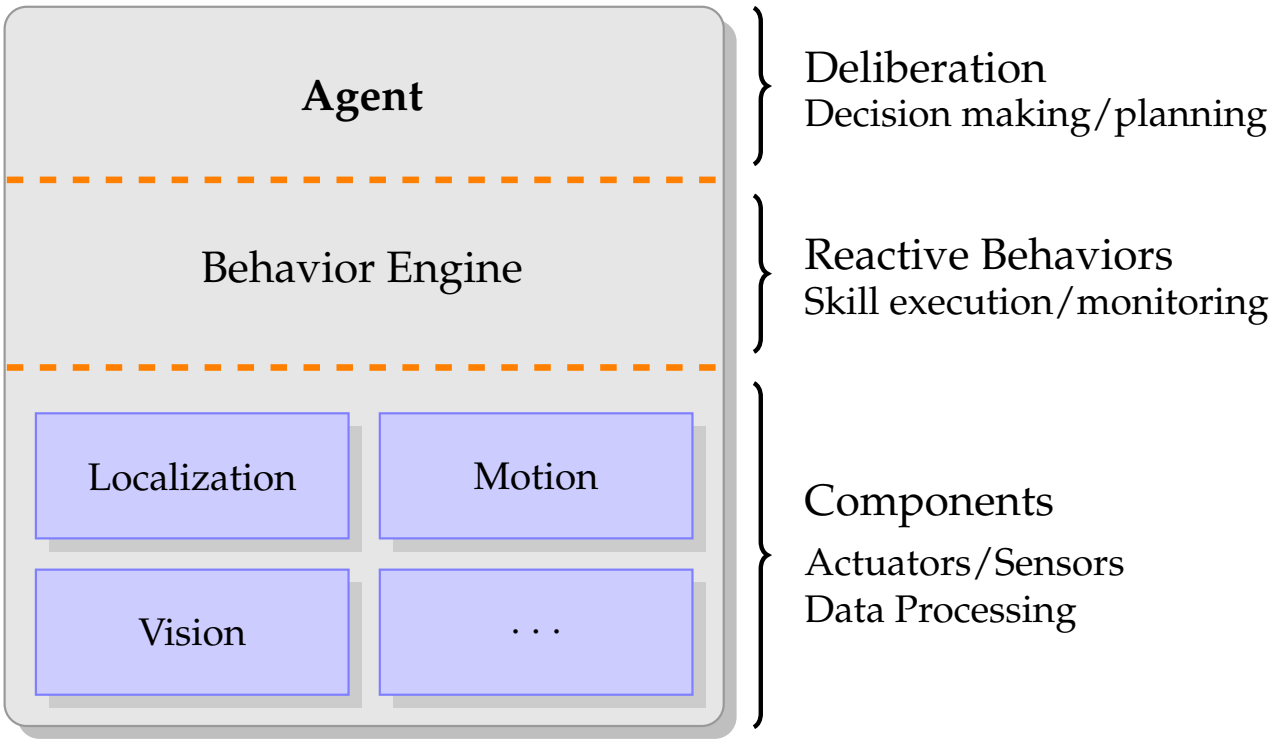


Transforms Graph

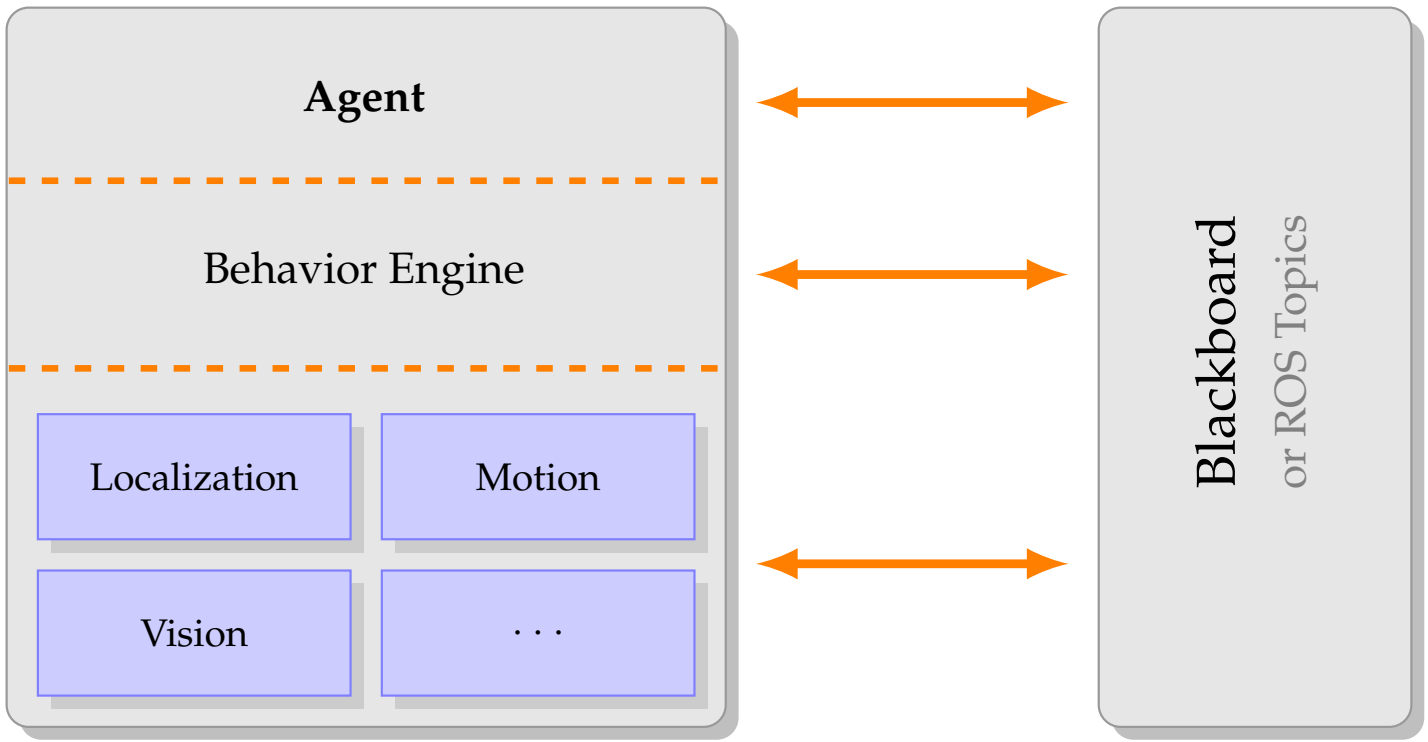
- Represent transforms as graph (tree)
- Walk the tree to calculate transform
- Performs time matching and interpolation
- API to transform points, poses etc.
- Static vs. dynamic transforms
- API from C++, Lua, and CLIPS



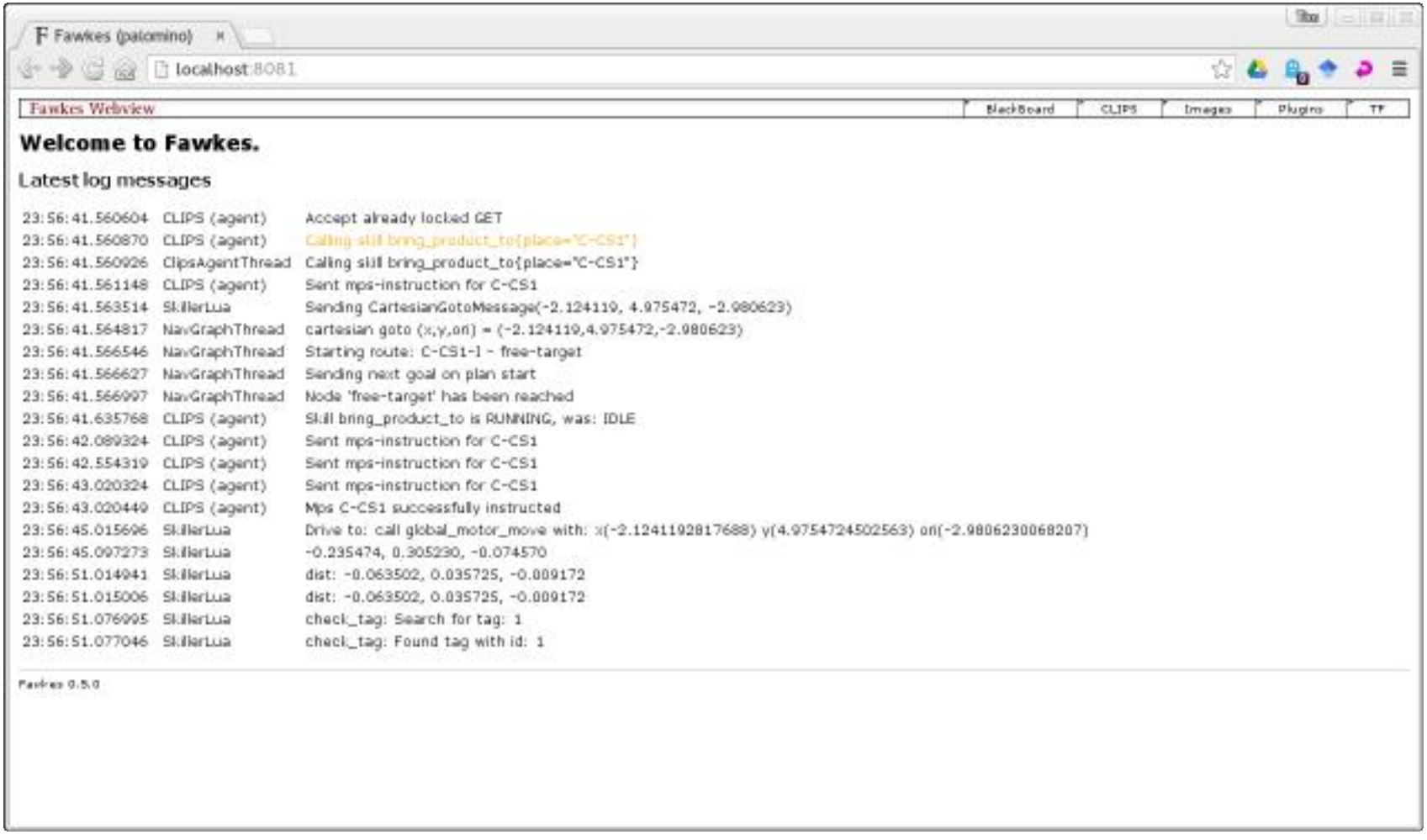
Behavioral Architecture



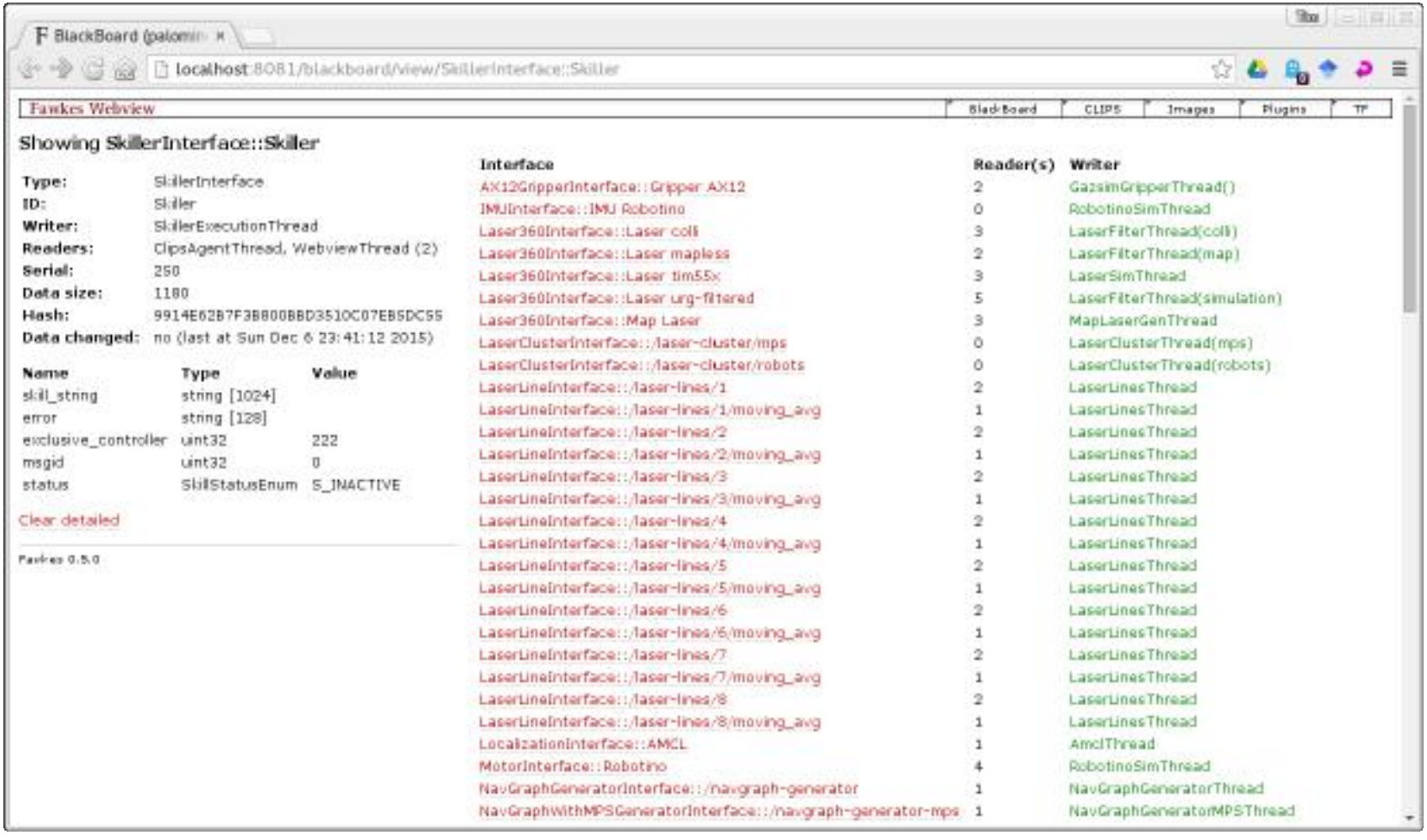
Behavioral Architecture



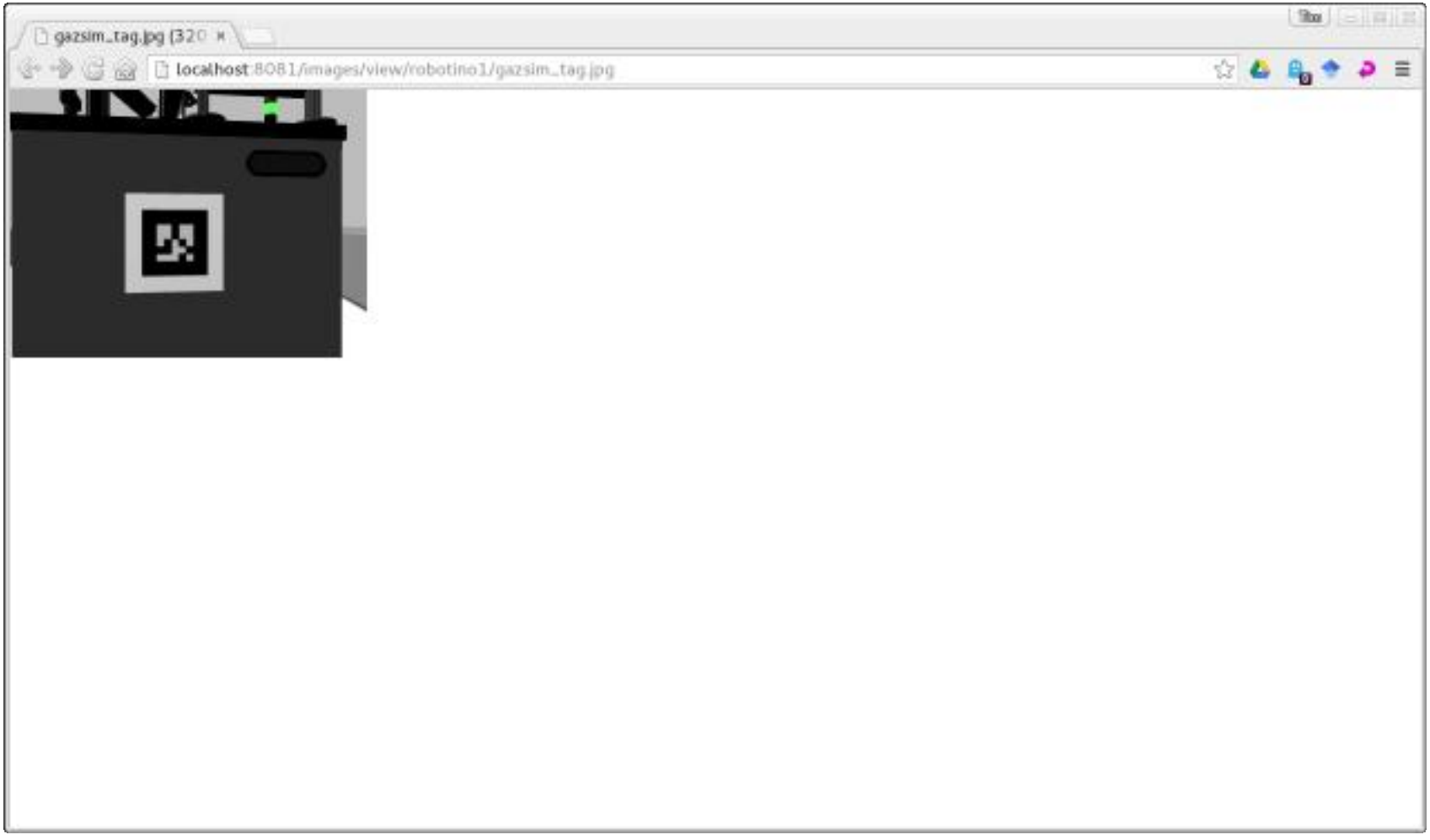
Webview


Fawkes (palomino)
localhost:8081
Fawkes Webview
BlackBoard
CLIPS
Images
Plugins
TF
Welcome to Fawkes.
Latest log messages
23:56:41.560604 CLIPS (agent) Accept already locked GET
23:56:41.560870 CLIPS (agent) Calling skill bring_product_to{place="C-CS1"}
23:56:41.560926 ClipsAgentThread Calling skill bring_product_to{place="C-CS1"}
23:56:41.561148 CLIPS (agent) Sent mps-instruction for C-CS1
23:56:41.563514 SkillerLua Sending CartesianGotoMessage(-2.124119, 4.975472, -2.980623)
23:56:41.564817 NavGraphThread cartesian goto (x,y,ori) = (-2.124119,4.975472,-2.980623)
23:56:41.566546 NavGraphThread Starting route: C-CS1-I - free-target
23:56:41.566627 NavGraphThread Sending next goal on plan start
23:56:41.566997 NavGraphThread Node 'free-target' has been reached
23:56:41.635768 CLIPS (agent) Skill bring_product_to is RUNNING, was: IDLE
23:56:42.089324 CLIPS (agent) Sent mps-instruction for C-CS1
23:56:42.554319 CLIPS (agent) Sent mps-instruction for C-CS1
23:56:43.020324 CLIPS (agent) Sent mps-instruction for C-CS1
23:56:43.020449 CLIPS (agent) Mps C-CS1 successfully instructed
23:56:45.015696 SkillerLua Drive to: call global_motor_move with: x(-2.1241192817688) y(4.9754724502563) ori(-2.9806230068207)
23:56:45.097273 SkillerLua -0.235474, 0.305290, -0.074570
23:56:51.014941 SkillerLua dist: -0.063502, 0.035725, -0.009172
23:56:51.015006 SkillerLua dist: -0.063502, 0.035725, -0.009172
23:56:51.076995 SkillerLua check_tag: Search for tag: 1
23:56:51.077046 SkillerLua check_tag: Found tag with id: 1
Fawkes 0.5.0

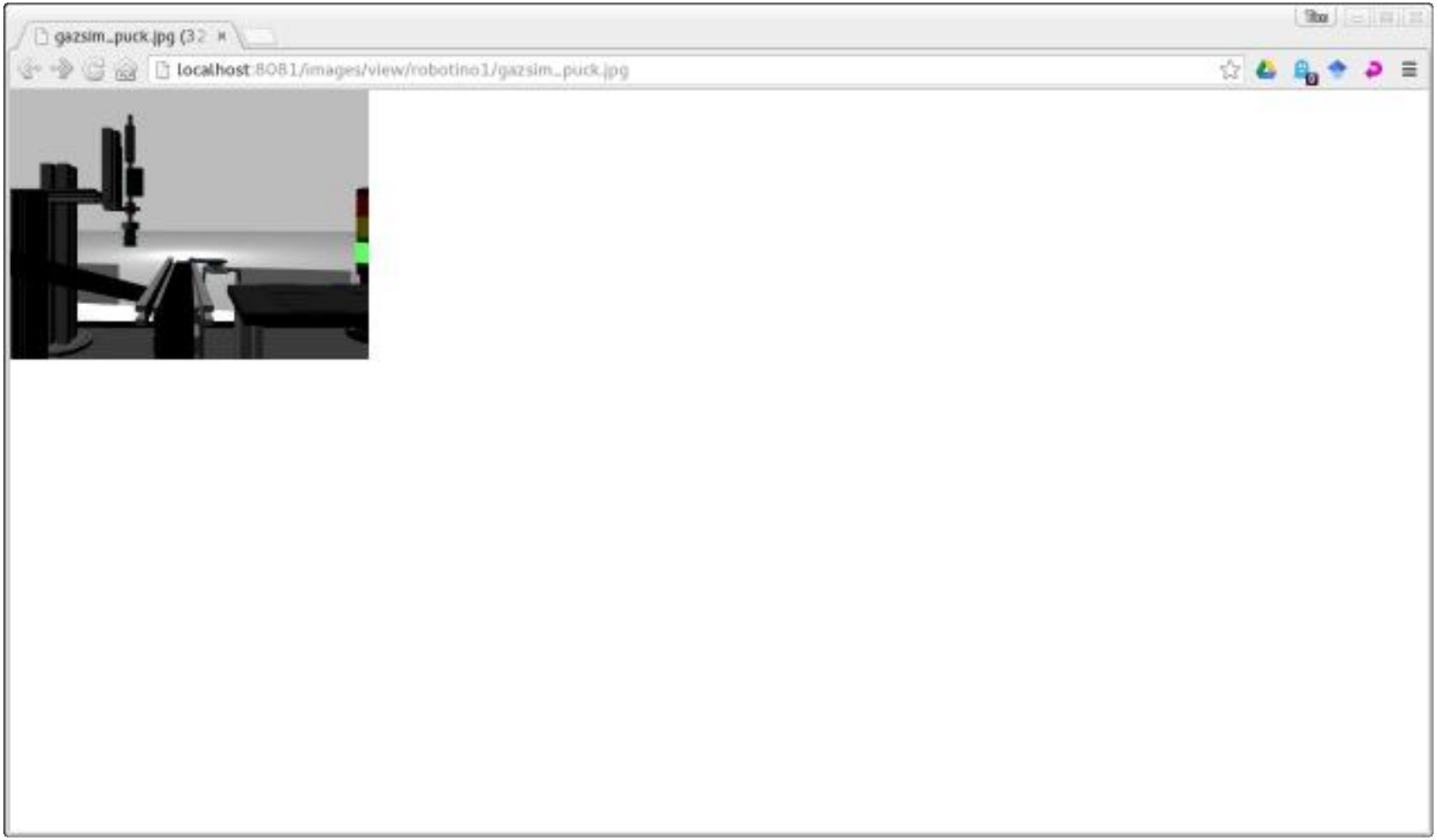
Webview


BlackBoard (palomin
localhost:8081/blackboard/view/SkillerInterface::Skiller
Fawkes Webview
BlackBoard
CLIPS
Images
Plugins
TF
Showing SkillerInterface::Skiller
Type: SkillerInterface
ID: Skiller
Writer: SkillerExecutionThread
Readers: ClipsAgentThread, WebviewThread (2)
Serial: 250
Data size: 1180
Hash: 9914E62B7F3B800BBD3510C07EB5DC55
Data changed: no (last at Sun Dec 6 23:41:12 2015)
Name Type Value
skill_string string [1024]
error string [128]
exclusive_controller uint32 222
msgid uint32 0
status SkillStatusEnum S_INACTIVE
Clear detailed
Fawkes 0.5.0
Interface Reader(s) Writer
AX12GripperInterface::Gripper AX12 2 GazsimGripperThread()
IMUInterface::IMU Robotino 0 RobotinoSimThread
Laser360Interface::Laser colli 3 LaserFilterThread(colli)
Laser360Interface::Laser mapless 2 LaserFilterThread(map)
Laser360Interface::Laser tim55x 3 LaserSimThread
Laser360Interface::Laser urg-filtered 5 LaserFilterThread(simulation)
Laser360Interface::Map Laser 3 MapLaserGenThread
LaserClusterInterface::/laser-cluster/mps 0 LaserClusterThread(mps)
LaserClusterInterface::/laser-cluster/robots 0 LaserClusterThread(robots)
LaserLineInterface::/laser-lines/1 2 LaserLinesThread
LaserLineInterface::/laser-lines/1/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/2 2 LaserLinesThread
LaserLineInterface::/laser-lines/2/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/3 2 LaserLinesThread
LaserLineInterface::/laser-lines/3/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/4 2 LaserLinesThread
LaserLineInterface::/laser-lines/4/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/5 2 LaserLinesThread
LaserLineInterface::/laser-lines/5/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/6 2 LaserLinesThread
LaserLineInterface::/laser-lines/6/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/7 2 LaserLinesThread
LaserLineInterface::/laser-lines/7/moving_avg 1 LaserLinesThread
LaserLineInterface::/laser-lines/8 2 LaserLinesThread
LaserLineInterface::/laser-lines/8/moving_avg 1 LaserLinesThread
LocalizationInterface::AMCL 1 AmclThread
MotorInterface::Robotino 4 RobotinoSimThread
NavGraphGeneratorInterface::/navgraph-generator 1 NavGraphGeneratorThread
NavGraphWithMPSGeneratorInterface::/navgraph-generator-mps 1 NavGraphGeneratorMPSThread

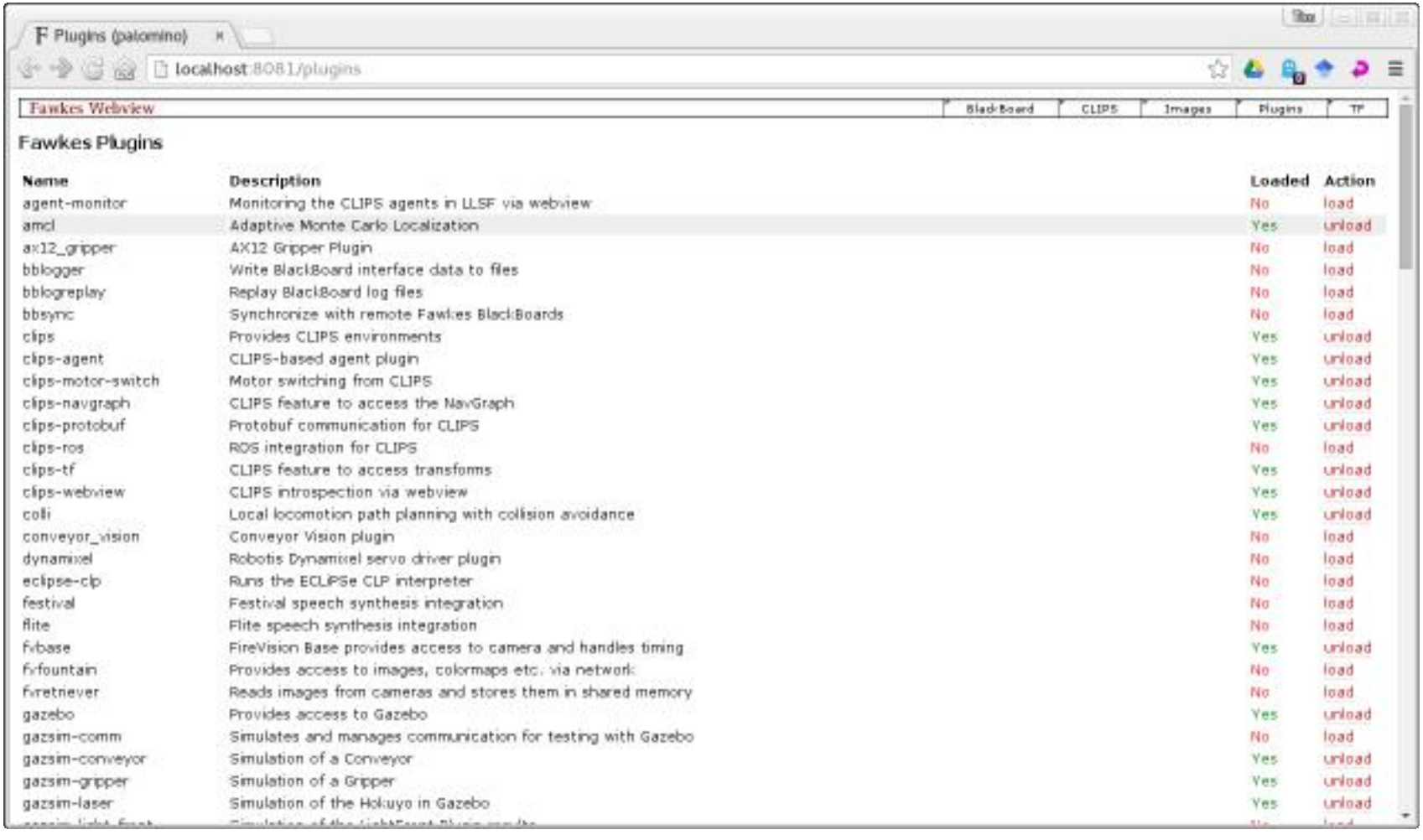
Webview


gazsim_tag.jpg (320
localhost:8081/images/view/robotino1/gazsim_tag.jpg

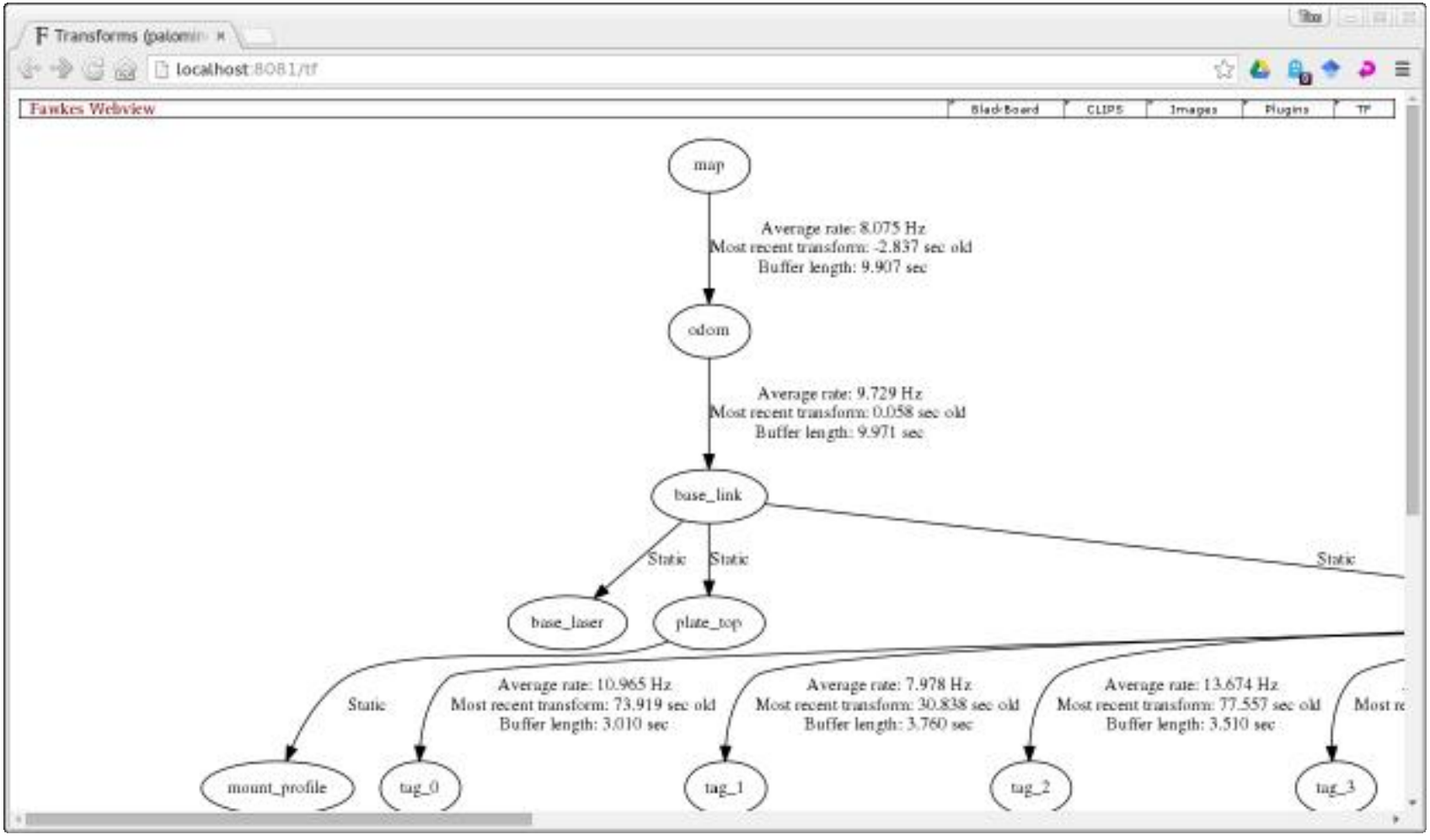
Webview


gazsim_puck.jpg (32
localhost:8081/images/view/robotino1/gazsim_puck.jpg

Webview


Plugins (palomino)
localhost:8081/plugins
Fawkes Webview
BlackBoard
CLIPS
Images
Plugins
TF
Fawkes Plugins
Name
Description
Loaded
Action
agent-monitor
Monitoring the CLIPS agents in LLSF via webview
No
load
amcl
Adaptive Monte Carlo Localization
Yes
unload
ax12_gripper
AX12 Gripper Plugin
No
load
bblogger
Write BlackBoard interface data to files
No
load
bblogreplay
Replay BlackBoard log files
No
load
bbsync
Synchronize with remote Fawkes BlackBoards
No
load
clips
Provides CLIPS environments
Yes
unload
clips-agent
CLIPS-based agent plugin
Yes
unload
clips-motor-switch
Motor switching from CLIPS
Yes
unload
clips-navgraph
CLIPS feature to access the NavGraph
Yes
unload
clips-protobuf
Protobuf communication for CLIPS
Yes
unload
clips-ros
ROS integration for CLIPS
No
load
clips-tf
CLIPS feature to access transforms
Yes
unload
clips-webview
CLIPS introspection via webview
Yes
unload
colli
Local locomotion path planning with collision avoidance
Yes
unload
conveyor_vision
Conveyor Vision plugin
No
load
dynamixel
Robotis Dynamixel servo driver plugin
No
load
eclipse-clp
Runs the ECLiPSe CLP interpreter
No
load
festival
Festival speech synthesis integration
No
load
flite
Flite speech synthesis integration
No
load
fvbase
FireVision Base provides access to camera and handles timing
Yes
unload
fvfountain
Provides access to images, colormaps etc. via network
No
load
fvretriever
Reads images from cameras and stores them in shared memory
No
load
gazebo
Provides access to Gazebo
Yes
unload
gazsim-comm
Simulates and manages communication for testing with Gazebo
No
load
gazsim-conveyor
Simulation of a Conveyor
Yes
unload
gazsim-gripper
Simulation of a Gripper
Yes
unload
gazsim-laser
Simulation of the Hokuyo in Gazebo
Yes
unload

Webview


Transforms (palomin
localhost:8081/tf
Fawkes Webview
Black-Board
CLIPS
Images
Plugins
TF
map
Average rate: 8.075 Hz
Most recent transform: -2.837 sec old
Buffer length: 9.907 sec
odom
Average rate: 9.729 Hz
Most recent transform: 0.058 sec old
Buffer length: 9.971 sec
base_link
Static
Static
Static
base_laser
plate_top
Static
Average rate: 10.965 Hz
Most recent transform: 73.919 sec old
Buffer length: 3.010 sec
Average rate: 7.978 Hz
Most recent transform: 30.838 sec old
Buffer length: 3.760 sec
Average rate: 13.674 Hz
Most recent transform: 77.557 sec old
Buffer length: 3.510 sec
mount_profile
tag_0
tag_1
tag_2
tag_3

Further Features

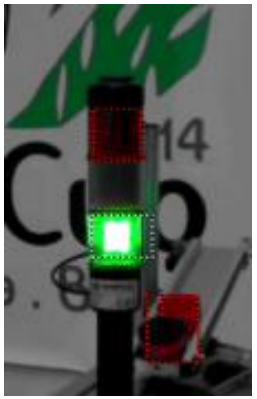
- Text and data logging facilities
- Configuration sub-system
- Network messaging infrastructure and discovery
- ROS integration
- Plugins for performance analysis (RRD, mongodb-log, ...)

$\Rightarrow$ Batteries included

Perception

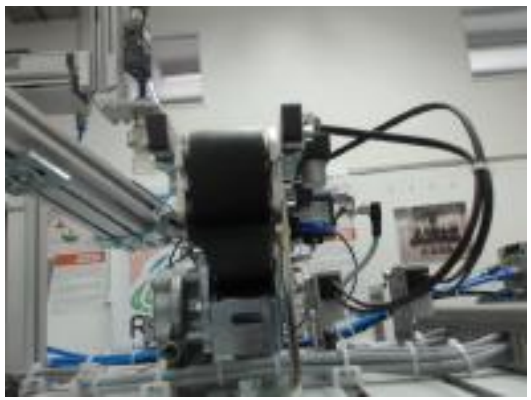
Machine Signal

- Recognize light signal on MPS
- Combine laser and image data to detect signal
- Robust to disturbances



Conveyor Detection

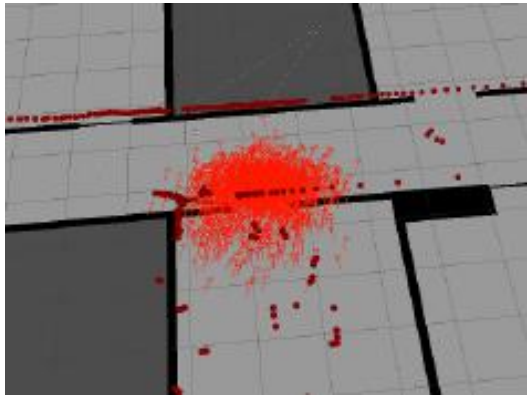
- Detect conveyor on MPS
- Based on point clouds from 3D vision
- Used for precise positioning



Navigation

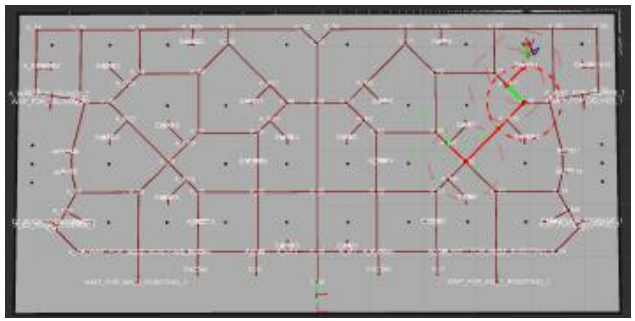
Self-localization

- Adaptive Monte Carlo Localization
- Particle filter on poses
- Works on 2D laser data



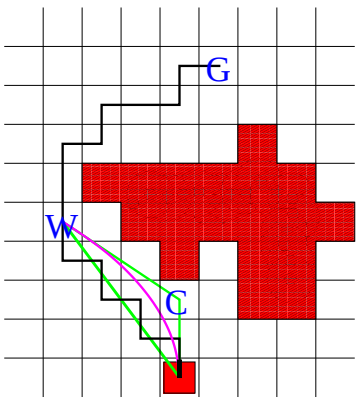
Global Path Planning

- Topological graph search
- Provides points of interest
- Statically collision-free paths



Local Path Planning

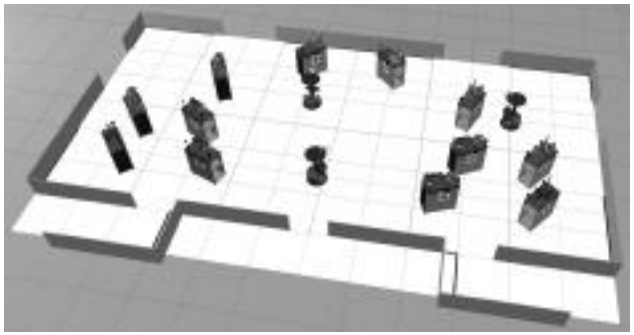
- Grid map approach on laser data
- Avoid dynamic and static obstacles
- Take global path as guide line



Simulation

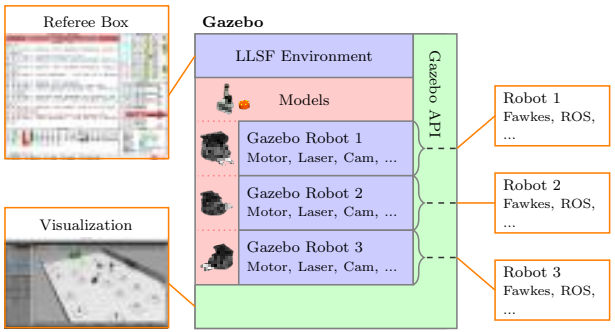
Gazebo-based Simulation

- Full 3D simulation with physics
- Based on well-known Gazebo simulator



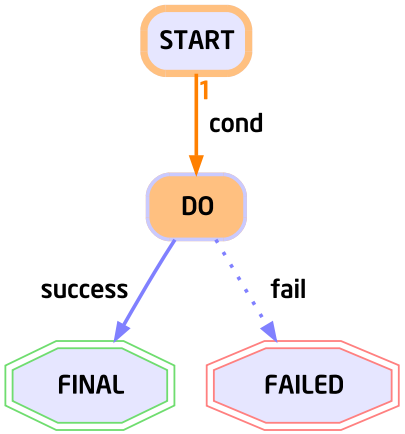
RCLL in Simulation

- Use referee box for environment agency
- Very close to the real game



Lua-based Behavior Engine

- Basic actions for reasoning layer
- Emphasize description over programming
- Allow programming where necessary
- Modeled using Hybrid State Machines
- Abstract low-level system
- Implemented for Fawkes and ROS
- Written in the Lua scripting language

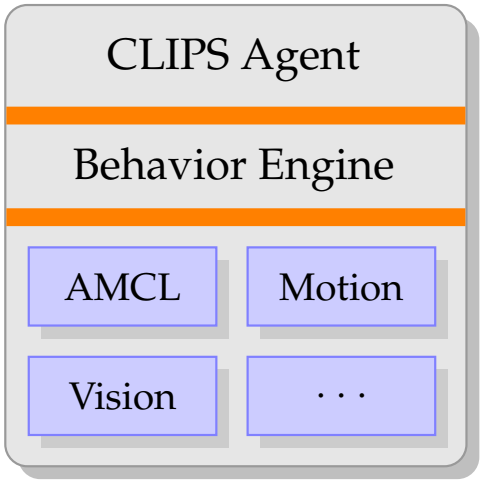


Variable table

x	5.2
y	4.3
error	
. . .	

Incremental Task-Level Reasoning

- Only commit to single step at a time
- Strategic behavior with coarse tasks
- Reason about current knowledge



- CLIPS rule-based system
- Efficient reasoning with many updates
- Distributed, local-scope, incremental


```
(defrule s1-t23-s0
  (state IDLE) (holding S1)
  (machine (mtype ?mt&M2\_3) (name ?n)
           (loaded-with $?lw&:(contains$ S0 ?lw)) )
  =>
  (assert (task-candidate goto ?n))
)
```

Conclusion and Questions

Fawkes as a versatile software framework is the foundation for the the publicly released Carologistics software stack.

- Hybrid blackboard middleware
- Massively multi-threaded software components
- Versatile Behavior Engine and reasoning agent
- Focus on integration with reasoning components
- RCLL software stack released as open source software

https://www.fawkesrobotics.org/
https://www.carologistics.org/