The Fawkes Robot Software Framework
Software Stack for the RoboCup Logistics League

Tim Niemueller
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 at a Glance

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

http://www.fawkesrobotics.org
1 Introduction

2 Fawkes Robot Software Framework

3 Software Components

4 Conclusion
1 Introduction

2 Fawkes Robot Software Framework

3 Software Components

4 Conclusion
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

![Diagram of Run-time Coordination]

**Pre Loop**
- Sensor/Vision
- Sensor Proc.
- World State
- Think
- Skill
- Act
- Act Exec
- Post Loop

**Concurrent Thread**
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
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
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
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 created by Fawkes main application
BlackBoard

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

Remote applications can access BlackBoard via network

Transactions (read/write)
Fawkes Threads access the BlackBoard via these Interfaces
Remote applications can access BlackBoard via network
Transactions (read/write)
Only one writer at a time
- Message passing as command channel
- Messages can only be sent from reader to writer
- Any number of messages in queue
- 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
Aspects

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
  ⇒ 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

<table>
<thead>
<tr>
<th>LaserLinesThread</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClockAspect</td>
</tr>
<tr>
<td>LoggingAspect</td>
</tr>
<tr>
<td>BlackBoardAspect</td>
</tr>
<tr>
<td>TransformAspect</td>
</tr>
<tr>
<td>PointCloudAspect</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
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

- 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

- 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

- 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

- 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
- 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

Agent

Behavior Engine

- Localization
- Motion
- Vision
- ...

Components
- Actuators/Sensors
- Data Processing

Reactive Behaviors
- Skill execution/monitoring

Deliberation
- Decision making/planning
Behavioral Architecture
Welcome to Fawkes.

Latest log messages

23:56:41.560504 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 SkillLua Sending CartesianGotoMessage(-2.124119, 4.975472, -2.980523)
23:56:41.564817 NavGraphThread cartesian goto (x,y,ori) = (-2.124119,4.975472,-2.980523)
23:56:41.566546 NavGraphThread Starting route: C-CS1-1 - free-target
23:56:41.566827 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
23:56:42.689324 CLIPS (agent) Sent mps-instruction for C-CS1
23:56:42.554329 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
23:56:45.015596 SkillLua Drive to: call global_motor_move with: x(-2.1241192817688) y(4.9754724502563) ori(-2.9806230069207)
23:56:45.097273 SkillLua -0.035474, 0.305230, -0.074570
23:56:51.014941 SkillLua dist: -0.063502, 0.035725, -0.009172
23:56:51.015006 SkillLua dist: -0.063502, 0.035725, -0.009172
23:56:51.076995 SkillLua check_tag: Search for tag: 1
23:56:51.077046 SkillLua check_tag: Found tag with id: 1
### Showing SkillerInterface::Skiller

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>skill_string</td>
<td>string [1024]</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td>string [128]</td>
<td></td>
</tr>
<tr>
<td>exclusive_controller</td>
<td>uint32</td>
<td>222</td>
</tr>
<tr>
<td>msgid</td>
<td>uint32</td>
<td>0</td>
</tr>
<tr>
<td>status</td>
<td>SkillStatusEnum</td>
<td>S_INACTIVE</td>
</tr>
</tbody>
</table>

### Interface

<table>
<thead>
<tr>
<th>Interface</th>
<th>Reader(s)</th>
<th>Writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX12GripperInterface::Gripper AX12</td>
<td>2</td>
<td>GazsimGripperThread()</td>
</tr>
<tr>
<td>IMUInterface::IMU Robotino</td>
<td>0</td>
<td>RobotinoSimThread</td>
</tr>
<tr>
<td>Laser360Interface::Laser colli</td>
<td>3</td>
<td>LaserFilterThread(Coll)</td>
</tr>
<tr>
<td>Laser360Interface::Laser mapless</td>
<td>2</td>
<td>LaserFilterThread(map)</td>
</tr>
<tr>
<td>Laser360Interface::Laser timer</td>
<td>5</td>
<td>LaserFilterThread</td>
</tr>
<tr>
<td>Laser360Interface::Laser uqg-filtered</td>
<td></td>
<td>LaserFilterThread(simulation)</td>
</tr>
<tr>
<td>Laser360Interface::Map Laser</td>
<td>3</td>
<td>MapLaserGeneratorThread</td>
</tr>
<tr>
<td>LaserClusterInterface::Laser-cluster/mps</td>
<td>0</td>
<td>LaserClusterThread(mp)</td>
</tr>
<tr>
<td>LaserClusterInterface::Laser-cluster/robots</td>
<td>0</td>
<td>LaserClusterThread(robots)</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/1</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/1/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/2</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/2/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/3</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/3/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/4</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/4/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/5</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/5/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/6</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/6/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/7</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/7/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/8</td>
<td>2</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LaserLineInterface::Laser-lines/8/moving_avg</td>
<td>1</td>
<td>LaserLinesThread</td>
</tr>
<tr>
<td>LocalizationInterface::AMCL</td>
<td>1</td>
<td>AMCLThread</td>
</tr>
<tr>
<td>MotorInterface::Robotino</td>
<td>4</td>
<td>RobotinoSimThread</td>
</tr>
<tr>
<td>NavGraphGeneratorInterface::navgraph-generator</td>
<td>1</td>
<td>NavGraphGeneratorThread</td>
</tr>
<tr>
<td>NavGraphWithMPSGeneratorInterface::navgraph-generator-mps</td>
<td>1</td>
<td>NavGraphGeneratorMPSThread</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Loaded</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>agent-monitor</td>
<td>Monitoring the CLIPS agents in LLSF via webview</td>
<td>No</td>
</tr>
<tr>
<td>amcl</td>
<td>Adaptive Monte Carlo Localization</td>
<td>Yes</td>
</tr>
<tr>
<td>ax12_gripper</td>
<td>AX12 Gripper Plugin</td>
<td>No</td>
</tr>
<tr>
<td>bblogger</td>
<td>Write BlackBoard interface data to files</td>
<td>No</td>
</tr>
<tr>
<td>bblogreplay</td>
<td>Replay BlackBoard log files</td>
<td>No</td>
</tr>
<tr>
<td>bbsync</td>
<td>Synchronize with remote Fawkes BlackBoards</td>
<td>No</td>
</tr>
<tr>
<td>clips</td>
<td>Provides CLIPS environments</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-agent</td>
<td>CLIPS-based agent plugin</td>
<td>No</td>
</tr>
<tr>
<td>clips-motor-switch</td>
<td>Motor switching from CLIPS</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-navgraph</td>
<td>CLIPS feature to access the NavGraph</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-protobuf</td>
<td>Protobuf communication for CLIPS</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-ros</td>
<td>ROS integration for CLIPS</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-tf</td>
<td>CLIPS feature to access transforms</td>
<td>Yes</td>
</tr>
<tr>
<td>clips-webview</td>
<td>CLIPS introspection via webview</td>
<td>Yes</td>
</tr>
<tr>
<td>colli</td>
<td>Local locomotion path planning with collision avoidance</td>
<td>Yes</td>
</tr>
<tr>
<td>conveyor_vision</td>
<td>Conveyor Vision plugin</td>
<td>No</td>
</tr>
<tr>
<td>dynamixel</td>
<td>Robotics Dynamixel servo driver plugin</td>
<td>No</td>
</tr>
<tr>
<td>eclipse-clp</td>
<td>Runs the ECLIPSa CLP interpreter</td>
<td>No</td>
</tr>
<tr>
<td>festival</td>
<td>Festival speech synthesis integration</td>
<td>No</td>
</tr>
<tr>
<td>flute</td>
<td>Flute speech synthesis integration</td>
<td>No</td>
</tr>
<tr>
<td>fubase</td>
<td>FireVision Base provides access to camera and handles timing</td>
<td>Yes</td>
</tr>
<tr>
<td>fyfountain</td>
<td>Provides access to images, colormaps etc., via network</td>
<td>No</td>
</tr>
<tr>
<td>fytreceiver</td>
<td>Reads images from cameras and stores them in shared memory</td>
<td>No</td>
</tr>
<tr>
<td>gazebo</td>
<td>Provides access to Gazebo</td>
<td>Yes</td>
</tr>
<tr>
<td>gazsim-cemm</td>
<td>Simulates and manages communication for testing with Gazebo</td>
<td>No</td>
</tr>
<tr>
<td>gazsim-conveyor</td>
<td>Simulation of a Conveyor</td>
<td>Yes</td>
</tr>
<tr>
<td>gazsim-gripper</td>
<td>Simulation of a Gripper</td>
<td>Yes</td>
</tr>
<tr>
<td>gazsim-laser</td>
<td>Simulation of the Hokuyo in Gazebo</td>
<td>Yes</td>
</tr>
<tr>
<td>gazebo_lightcone</td>
<td>Simulation of the Lightcone Gazebo</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Further Features

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

2 Fawkes Robot Software Framework

3 Software Components

4 Conclusion
Laser Lines
Conveyor Vision
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
- Haar cascades trained on conveyor
- (not in this winter school)

Day 2 – morning
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

Day 2 – afternoon
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

Day 3 – full day
Behavioral Architecture

Agent

Behavior Engine

Localisation
Motion
Vision
...

Components
Actuators/Sensors
Data Processing

Reactive Behaviors
Skill execution/monitoring

Deliberation
Decision making/planning
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

Day 4 and 5 combined with agent

Variable table

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>5.2</td>
</tr>
<tr>
<td>y</td>
<td>4.3</td>
</tr>
<tr>
<td>error</td>
<td>3.3</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
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))
)
Fawkes as a versatile software framework is the foundation for 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/