ILIAD

Intra-Logistics with Integrated Automatic Deployment:
Safe and Scalable Fleets in Shared Spaces

H2020-ICT-2016-2017
Grant agreement no: 732737

DELIVERABLE 7.2
System architecture

Due date: month 9 (September 2017)
Deliverable type: O
Lead beneficiary: UoL

Dissemination Level: PUBLIC

Main author: Marc Hanheide (UOL)

Contents

List of Figures

Executive Summary

This document covers the design and implementation plan for the ILIAD system architecture. It is seen as an executive summary of the approach, with links to further online resources, that will continuously be updated as the development progresses, and serve as documentation for the ILIAD development teams internally, as well as providing essential documentation on usage and development within the ILIAD software distribution. The deliverable is of type "O" ("other"), due to its nature of being mostly a collection of online resources and the corresponding infrastructure that generates and maintains them, as well as a dedicated software release of a first instantiation of the ILIAD system architecture.

1 Introduction

This document accompanies the efforts in ILIAD to devise a system architecture and software development and deployment environment. ILIAD's objectives relating to this effort are set out in WP7 below.

Objectives addressed (from Description of Action)

- Thorough and dependable integration of all software systems, ensuring reliable systems for long-term application in intra-logistics applications
- A modular and easily maintainable software architecture, with the double purpose of making development within the project more efficient while also maximising the impact and uptake of its outcomes.

2 Architectural paradigms

ILIAD's system architecture is inherently a component-based architecture, built around the principles of re-usability and low coupling between components. Functional requirements alongside these two main principles are the guiding paradigms in the development of the architecture, leading to 3-layered functional architecture described in Sec. 2.1, and development and deployment process facilitating distributed and component-based implementation, with rigid review and testing procedures in place, build upon continuation integration.

The infrastructure and software frameworks are largely adopted from previous projects (STRANDS, SAUNA, Semantic Robots, SPENCER), and mostly based on community-available middlewares and other tools, almost exclusively ROS.[1] As a consequence, at the core of ILIAD sits a distributed processing model facilitated by the ROS middleware. The consortium has agreed that the main target platform for the development (at least for the first two years of the project, with re-assessment of the decision in Y3) is Ubuntu Linux Xenial (16.04 LTS, 64-bit) with ROS version "Kinetic".

2.1 Functional Architecture

In general, the distributed system architecture of the fleet is designed around functional layers:

[1] http://www.ros.org

Robot Abilities (on-board control)

At the bottom, an *ability level* encapsulates the basic robot functionalities like localisation, individual reactive navigation behaviours, pick and place operations, etc, running on the processing units of the individual robots in the fleet. This facilitates low-latency, closed-loop control of individual lower-level abilities of the robots, and is based entirely on ROS components, deployed locally to the robots in the fleet.

Shared Knowledge Management

In the functional architecture, on top of the ability level, we designed a shared knowledge and consolidation layer that encompasses the information fusion and sharing abilities of the robotic fleet, like mapping, perception consolidation, and data sharing. These comprise services that run on cloud or site-local computing platforms, following a Software as a Service (SaaS) paradigm to facilitate dissemination and exploitation through accessible APIs of the generic implementations. In the first implementation of the layer, communication is mainly facilitated via ROS services, but in the course of the project these can be more and more opened up to publicly accessible web services where appropriate.

Fleet Management & Coordination

At the top of this conceptual architecture, the distributed aspects of fleet management (WP5) will be implemented. This layer comprises fleet-level planning and scheduling, consulting warehouse operations-level systems and expertise for selecting tasks to be accomplished, and coordinates the fleet given constraints provided from the shared knowledge level.

2.2 Architecture Diagram

The overall intended architecture, designed based on the requirement analysis (see D7.1), is presented in Figure 1. It is based on the functional model outlined in Section 2.1. It captured the main directions of data flow on a conceptual level and to some extent abstracts from actual software packages. The currently released software packages (at M9) are captured in Figure 2. While Figure 1 captures the functional architecture that guides the development and integration in ILIAD, Figure 2 is an automatically generated snapshot of the status of software releases in ILIAD, and their run-time dependencies. The packages presented in the latter figure are available as part of the ILIAD distribution to any outside collaborators, simply as Ubuntu packages. Figure 2 is a snapshot of the released packages in M9 of ILIAD. This figure is automatically updated as part of the automated software documentation process and an updated version is always available at https://github.com/iliad-project/iliad-public/wiki/repos.

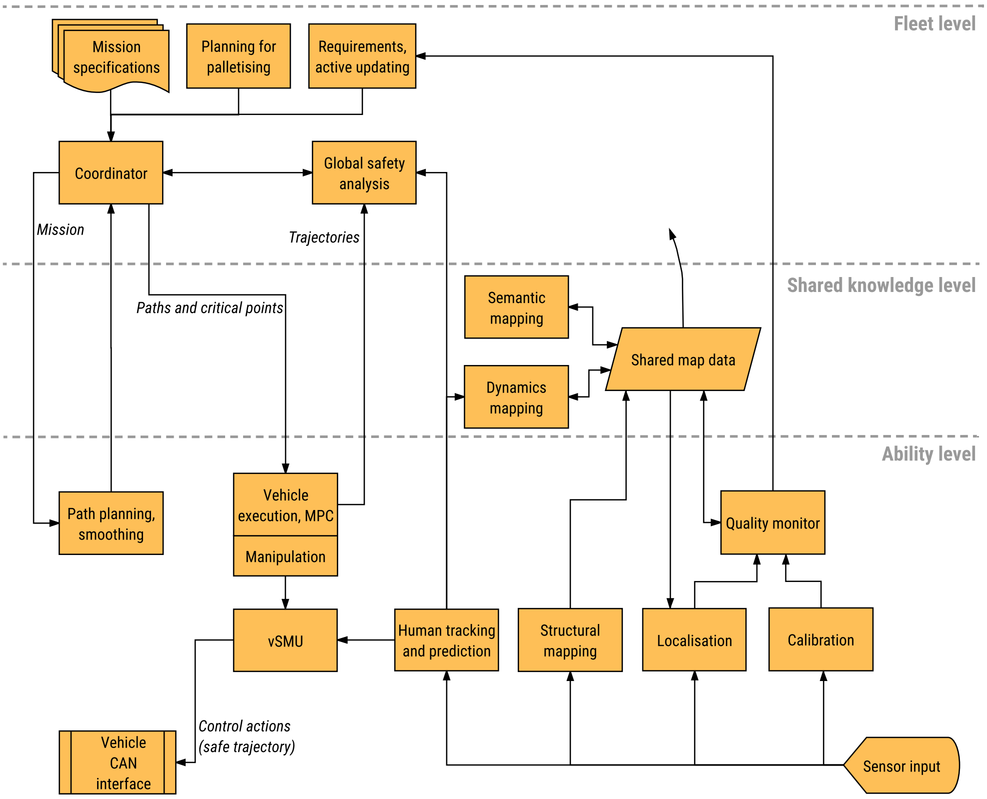


Figure 1: Overview of architecture.

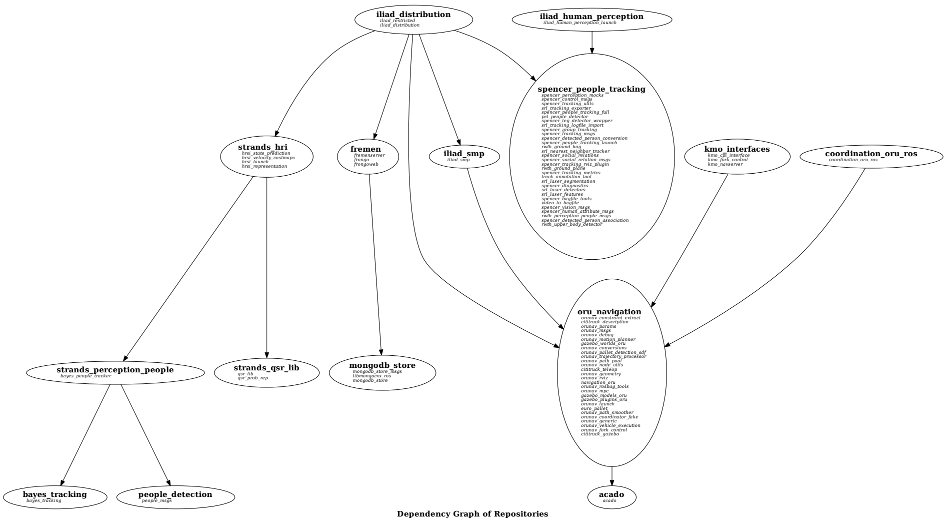
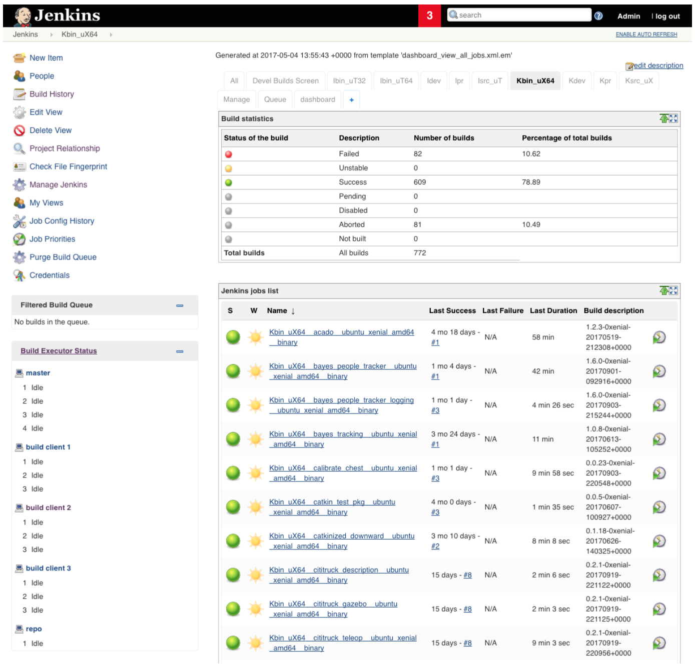


Figure 2: Static run-time dependencies between ILIAD repositories and packages (snapshot of released packages for MS1).



Generated at 2017-05-04 13:55:43 +0000 from template 'dashboard_view_all_jobs.xml.em'

Build statistics

Status of the build	Description	Number of builds	Percentage of total builds
🔴	Failed	82	10.62
🟡	Unstable	0	
🟢	Success	609	78.89
●	Pending	0	
●	Disabled	0	
●	Aborted	81	10.49
●	Not built	0	
Total builds	All builds	772	

Jenkins jobs list

S	W	Name ↓	Last Success	Last Failure	Last Duration	Build description
🟢	☀	Kbin_uX64__acado__ubuntu_xenial_amd64__binary	4 mo 18 days - #1	N/A	58 min	1.2.3-0xenial-20170519-212308+0000
🟢	☀	Kbin_uX64__bayes_people_tracker__ubuntu_xenial_amd64__binary	1 mo 4 days - #1	N/A	42 min	1.6.0-0xenial-20170901-092916+0000
🟢	☀	Kbin_uX64__bayes_people_tracker_logging__ubuntu_xenial_amd64__binary	1 mo 1 day - #3	N/A	4 min 26 sec	1.6.0-0xenial-20170903-215244+0000
🟢	☀	Kbin_uX64__bayes_tracking__ubuntu_xenial_amd64__binary	3 mo 24 days - #1	N/A	11 min	1.0.8-0xenial-20170613-105252+0000
🟢	☀	Kbin_uX64__calibrate_chest__ubuntu_xenial_amd64__binary	1 mo 1 day - #3	N/A	9 min 58 sec	0.0.23-0xenial-20170903-220548+0000
🟢	☀	Kbin_uX64__catkin_test_pkg__ubuntu_xenial_amd64__binary	4 mo 0 days - #3	N/A	1 min 35 sec	0.0.5-0xenial-20170607-100927+0000
🟢	☀	Kbin_uX64__catkinized_downward__ubuntu_xenial_amd64__binary	3 mo 10 days - #2	N/A	8 min 8 sec	0.1.18-0xenial-20170626-140325+0000
🟢	☀	Kbin_uX64__cititruck_description__ubuntu_xenial_amd64__binary	15 days - #8	N/A	2 min 6 sec	0.2.1-0xenial-20170919-221122+0000
🟢	☀	Kbin_uX64__cititruck_gazebo__ubuntu_xenial_amd64__binary	15 days - #8	N/A	2 min 3 sec	0.2.1-0xenial-20170919-221125+0000
🟢	☀	Kbin_uX64__cititruck_teleop__ubuntu_xenial_amd64__binary	15 days - #8	N/A	9 min 3 sec	0.2.1-0xenial-20170919-220956+0000

Figure 3: Screenshot of Jenkins CI server of ILIAD, running at https://lcas.lincoln.ac.uk/buildfarm/

3 Software Development Process and Integration Infrastructure

Software development in ILIAD is supported by a continuous integration (CI) server and a cloud-based build farm, serving release and deployment functionality, to allow ILIAD to build binary packages of released software, easing installation for third-parties and members of the consortium. The CI server is based on Jenkins[2] and is based on infrastructure originally developed in the STRANDS FP7 project (see screenshot in Figure 3).

3.1 Development Work Flow and Code Reviews

In order to ensure high software quality and ease of software deployment (also to maximise dissemination and uptake of ILIAD's outcomes), the Continuous Integration and deployment paradigm is facilitated through a build farm infrastructure that is committed to ILIAD by UoL (see above).

Within the project, we have agreed to use a git-based workflow, collaborating on software repositories stored publicly at http://github.com/ or other public git-based repositories, when possible, or at partner-local repository servers, when required to maintain intellectual property rights.

The general workflow follows the "fork and pull" model [1], which is well integrated with continuous integration. The general workflow is captured in Figure 4. ILIAD has designated named *maintainers* for individual software packages, who are responsible for code review and assessment of pull/merge requests. They are supported by the CI server, which will compile and unit-test any opened requests to facilitate the assessment prior to any merges to the master code base. Releases are only ever tagged on the master branch

[2] https://jenkins.io/

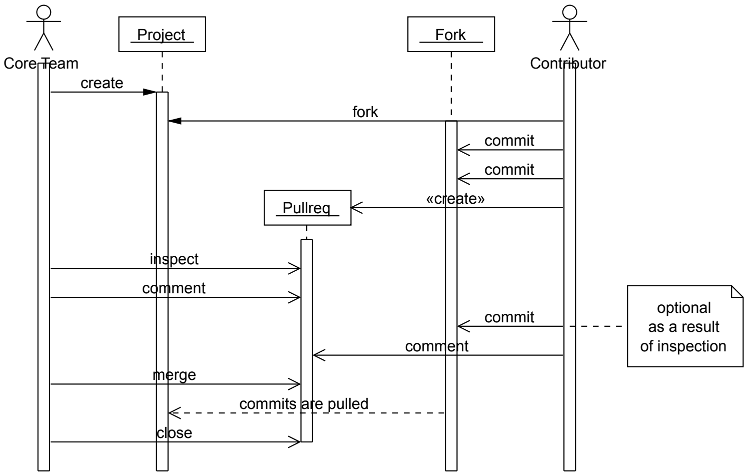


Figure 4: Overview of the fork and pull model used for code review and integration work flow in ILIAD (taken from [1]).

and undergo a dedicated review process by the technical board before accepted into the ILIAD distribution.

3.2 The ILIAD software distribution

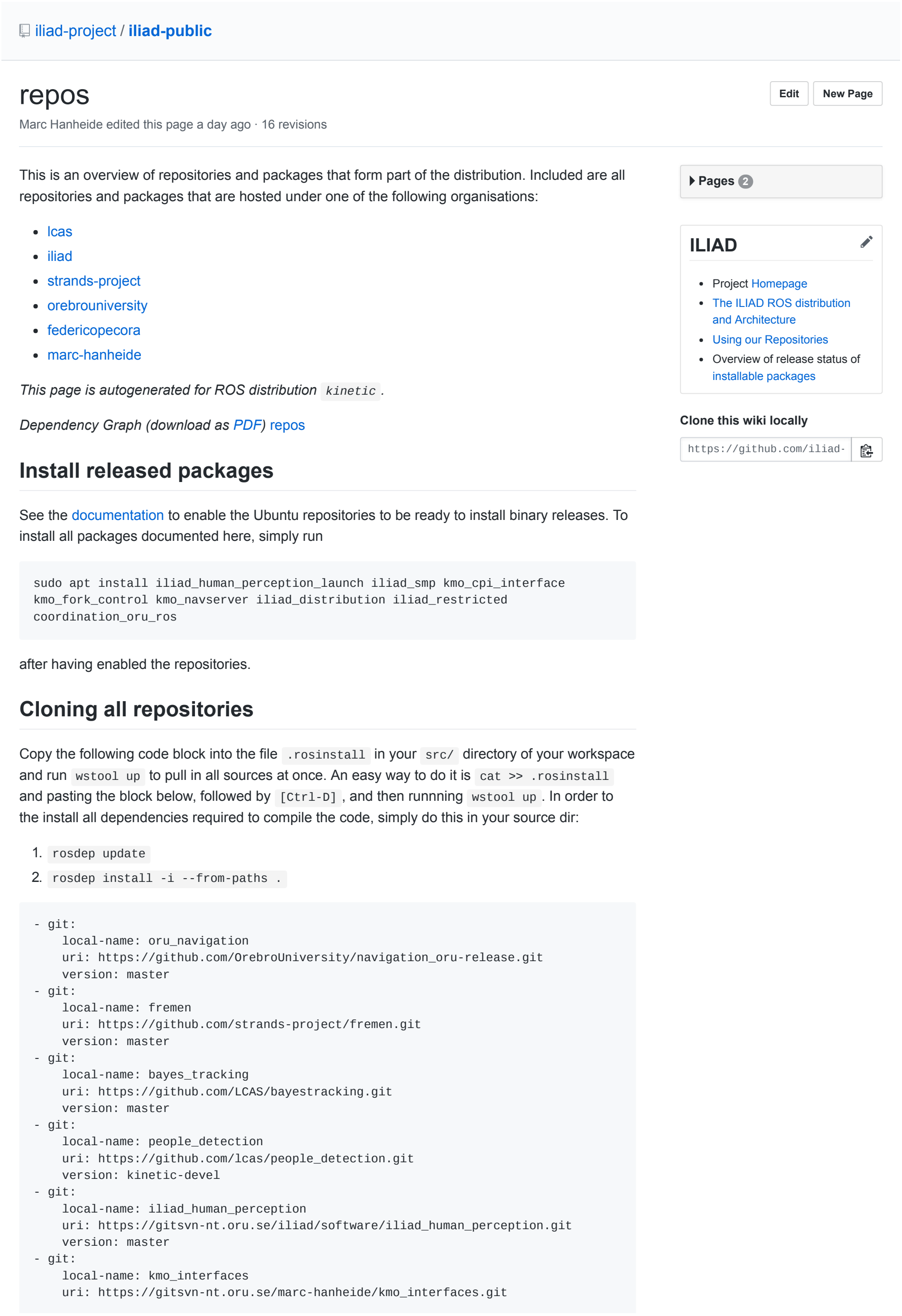
As referred to already earlier, ILIAD develops a dedicated software distribution, based mainly on ROS, and released both as source and easily installable binary packages. This is to ease the development and deployment processes and to maximise impact by enabling the community to adopt ILIAD software quickly and painlessly. As part of this D7.2, the infrastructure has been developed to maintain and document this software distribution using the following resources which are accessible publicly:

- An automatically maintained overview of all software packages of ILIAD, with instructions for development and installation: https://github.com/iliad-project/iliad-public/wiki/repos
- A Ubuntu software repository, with a three step guide to enable it in any Ubuntu-based computer system: https://github.com/LCAS/rosdistro/wiki
- A fully OSRF[3]-compatible release process, allowing release of ILIAD packages also into the ROS universe.

[3] https://www.osrfoundation.org/

A Snapshot of Distribution Documentation

This appendix shows a snapshot of the documentation available at the public-facing wiki for the ILIAD software distribution. See https://github.com/iliad-project/iliad-public/wiki/repos for an up-to-date version.



repos

Marc Hanheide edited this page a day ago · 16 revisions

This is an overview of repositories and packages that form part of the distribution. Included are all repositories and packages that are hosted under one of the following organisations:

- lcas
- iliad
- strands-project
- orebrouniversity
- federicopecora
- marc-hanheide

This page is autogenerated for ROS distribution `kinetic`.

Dependency Graph (download as PDF) repos

Install released packages

See the documentation to enable the Ubuntu repositories to be ready to install binary releases. To install all packages documented here, simply run

```
sudo apt install iliad_human_perception_launch iliad_smp kmo_cpi_interface
kmo_fork_control kmo_navserver iliad_distribution iliad_restricted
coordination_oru_ros
```

after having enabled the repositories.

Cloning all repositories

Copy the following code block into the file `.rosinstall` in your `src/` directory of your workspace and run `wstool up` to pull in all sources at once. An easy way to do it is `cat >> .rosinstall` and pasting the block below, followed by `[Ctrl-D]`, and then runnning `wstool up`. In order to the install all dependencies required to compile the code, simply do this in your source dir:

1. `rosdep update`
2. `rosdep install -i --from-paths .`

```
- git:
    local-name: oru_navigation
    uri: https://github.com/OrebroUniversity/navigation_oru-release.git
    version: master
- git:
    local-name: fremen
    uri: https://github.com/strands-project/fremen.git
    version: master
- git:
    local-name: bayes_tracking
    uri: https://github.com/LCAS/bayestracking.git
    version: master
- git:
    local-name: people_detection
    uri: https://github.com/lcas/people_detection.git
    version: kinetic-devel
- git:
    local-name: iliad_human_perception
    uri: https://gitsvn-nt.oru.se/iliad/software/iliad_human_perception.git
    version: master
- git:
    local-name: kmo_interfaces
    uri: https://gitsvn-nt.oru.se/marc-hanheide/kmo_interfaces.git
```

ILIAD

- Project Homepage
- The ILIAD ROS distribution and Architecture
- Using our Repositories
- Overview of release status of installable packages

Clone this wiki locally

continued:

```
    version: master
- git:
    local-name: iliad_distribution
    uri: https://gitsvn-nt.oru.se/iliad/software/iliad_metapackage.git
    version: master
- git:
    local-name: spencer_people_tracking
    uri: https://github.com/lcas/spencer_people_tracking.git
    version: master
- git:
    local-name: coordination_oru_ros
    uri: https://github.com/FedericoPecora/coordination_oru_ros.git
    version: master
- git:
    local-name: strands_hri
    uri: https://github.com/strands-project/strands_hri.git
    version: kinetic-devel
- git:
    local-name: strands_qsr_lib
    uri: https://github.com/strands-project/strands_qsr_lib.git
    version: master
- git:
    local-name: acado
    uri: https://github.com/LCAS/acado.git
    version: stable
- git:
    local-name: strands_perception_people
    uri: https://github.com/strands-project/strands_perception_people.git
    version: kinetic-devel
- git:
    local-name: mongodb_store
    uri: https://github.com/strands-project/mongodb_store.git
    version: kinetic-devel
```

oru_navigation

Source Code: https://github.com/OrebroUniversity/navigation_oru-release.git (branch: master)

`rosinstall` definition:

```
- git:
    local-name: oru_navigation
    uri: https://github.com/OrebroUniversity/navigation_oru-release.git
    version: master
```

included packages:

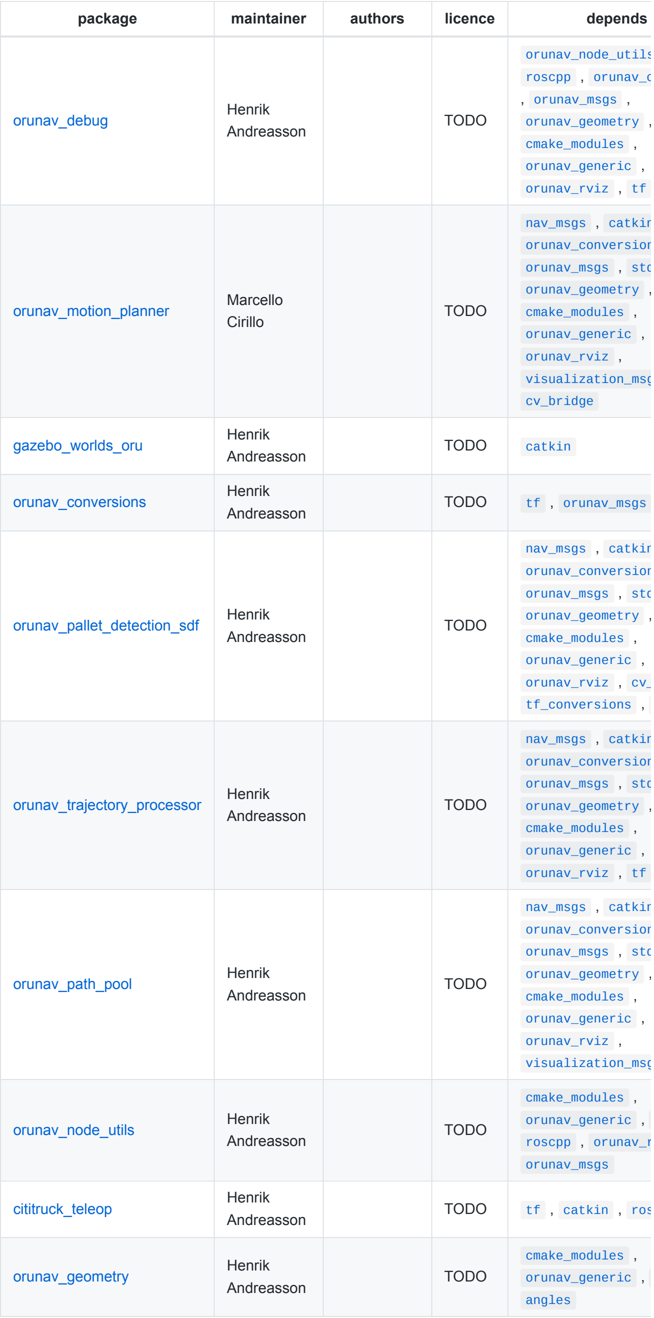
package	maintainer	authors	licence	depends

orunav_constraint_extract	Henrik Andreasson		TODO	`nav_msgs` , `catki`
				`orunav_conversio`
				`orunav_msgs` , `st`
				`orunav_geometry` ,
				`cmake_modules` ,
				`orunav_generic` ,
				`orunav_rviz` , `tf`

cititruck_description	Henrik Andreasson		TODO	`catkin`
orunav_params	Henrik Andreasson		TODO	`catkin`

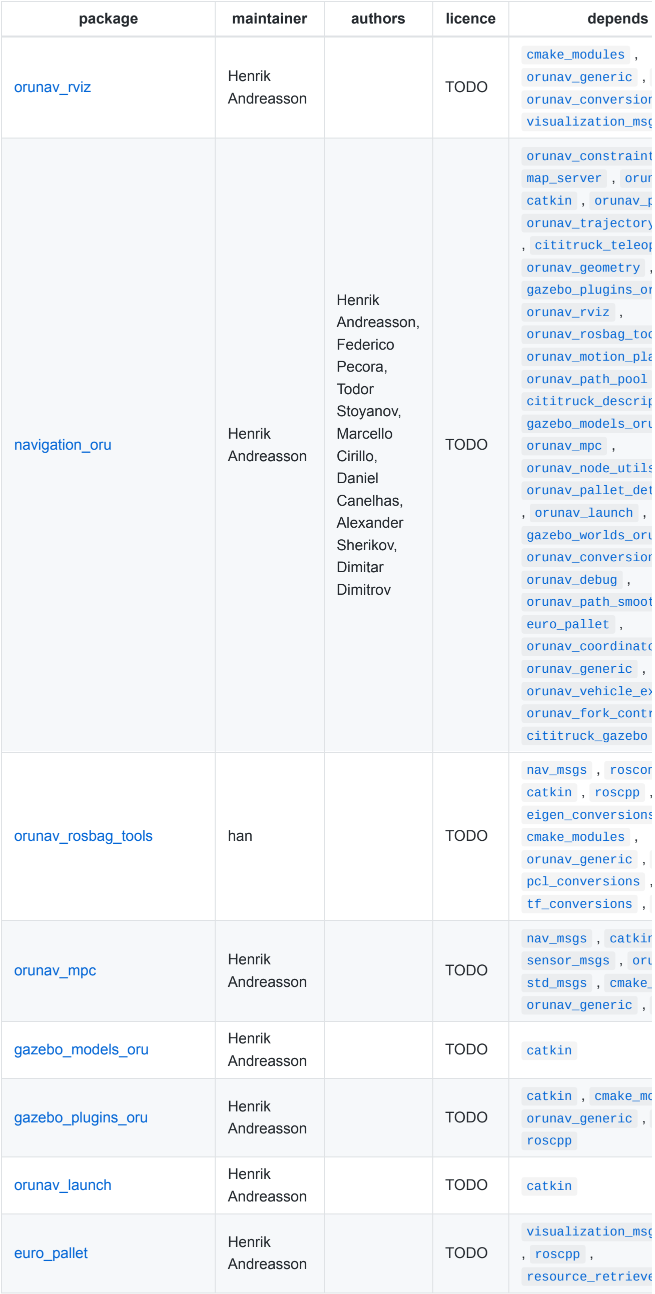
orunav_msgs	Henrik Andreasson, Federico Pecora		TODO	`nav_msgs` , `catki`
				`message_generati`
				`geometry_msgs` ,
				`message_runtime` ,

continued:



package	maintainer	authors	licence	depends
orunav_debug	Henrik Andreasson		TODO	orunav_node_utils roscpp , orunav_c , orunav_msgs , orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , tf
orunav_motion_planner	Marcello Cirillo		TODO	nav_msgs , catkin orunav_conversion orunav_msgs , std orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , visualization_msg cv_bridge
gazebo_worlds_oru	Henrik Andreasson		TODO	catkin
orunav_conversions	Henrik Andreasson		TODO	tf , orunav_msgs
orunav_pallet_detection_sdf	Henrik Andreasson		TODO	nav_msgs , catkin orunav_conversion orunav_msgs , std orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , cv_ tf_conversions ,
orunav_trajectory_processor	Henrik Andreasson		TODO	nav_msgs , catkin orunav_conversion orunav_msgs , std orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , tf
orunav_path_pool	Henrik Andreasson		TODO	nav_msgs , catkin orunav_conversion orunav_msgs , std orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , visualization_msg
orunav_node_utils	Henrik Andreasson		TODO	cmake_modules , orunav_generic , roscpp , orunav_r orunav_msgs
cititruck_teleop	Henrik Andreasson		TODO	tf , catkin , ros
orunav_geometry	Henrik Andreasson		TODO	cmake_modules , orunav_generic , angles

continued:



package	maintainer	authors	licence	depends
orunav_rviz	Henrik Andreasson		TODO	cmake_modules , orunav_generic , orunav_conversior visualization_msg
navigation_oru	Henrik Andreasson	Henrik Andreasson, Federico Pecora, Todor Stoyanov, Marcello Cirillo, Daniel Canelhas, Alexander Sherikov, Dimitar Dimitrov	TODO	orunav_constraint map_server , orur catkin , orunav_p orunav_trajectory , cititruck_teleop orunav_geometry , gazebo_plugins_or orunav_rviz , orunav_rosbag_too orunav_motion_pla orunav_path_pool cititruck_descrip gazebo_models_oru orunav_mpc , orunav_node_utils orunav_pallet_det , orunav_launch , gazebo_worlds_oru orunav_conversior orunav_debug , orunav_path_smoot euro_pallet , orunav_coordinato orunav_generic , orunav_vehicle_ex orunav_fork_contr cititruck_gazebo
orunav_rosbag_tools	han		TODO	nav_msgs , roscor catkin , roscpp , eigen_conversions cmake_modules , orunav_generic , pcl_conversions , tf_conversions ,
orunav_mpc	Henrik Andreasson		TODO	nav_msgs , catkir sensor_msgs , oru std_msgs , cmake_ orunav_generic ,
gazebo_models_oru	Henrik Andreasson		TODO	catkin
gazebo_plugins_oru	Henrik Andreasson		TODO	catkin , cmake_mc orunav_generic , roscpp
orunav_launch	Henrik Andreasson		TODO	catkin
euro_pallet	Henrik Andreasson		TODO	visualization_msg , roscpp , resource_retrieve

continued:

package	maintainer	authors	licence	depends
orunav_path_smoother	Henrik Andreasson		TODO	nav_msgs , orunav_constraint orunav_trajectory , catkin , roscpp orunav_conversior orunav_msgs , stc orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , aca
orunav_coordinator_fake	Henrik Andreasson		TODO	nav_msgs , catkir orunav_conversior orunav_msgs , stc orunav_geometry , cmake_modules , orunav_generic , orunav_rviz , visualization_msg
orunav_generic	Henrik Andreasson		TODO	cmake_modules , c angles
orunav_vehicle_execution	Henrik Andreasson, Federico Pecora		TODO	orunav_constraint orunav_trajectory , catkin , sensor_ orunav_msgs , stc orunav_geometry , cmake_modules , c , nav_msgs , orunav_node_utils laser_geometry , roscpp , orunav_c , orunav_generic ,
orunav_fork_control	Henrik Andreasson		TODO	catkin , roscpp , orunav_msgs , stc cmake_modules , orunav_generic , orunav_rviz , visualization_msg
cititruck_gazebo	Henrik Andreasson		TODO	catkin

fremen

Source Code: https://github.com/strands-project/fremen.git (branch: master)

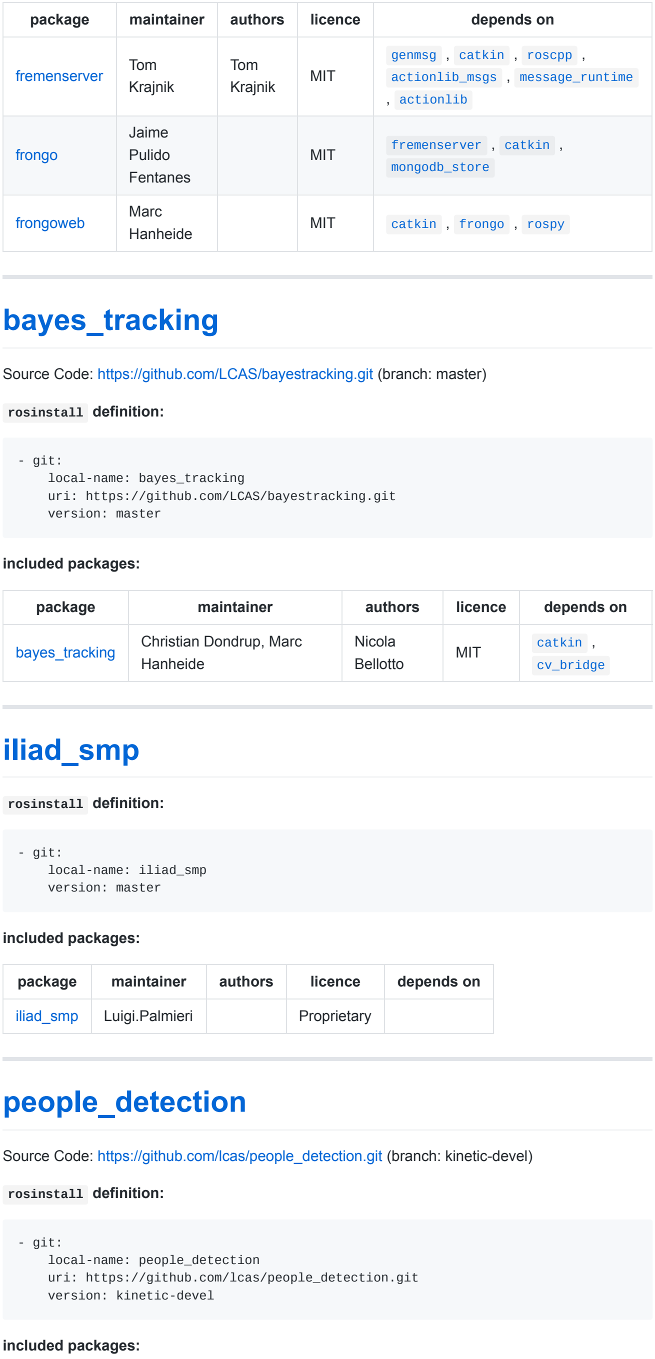
`rosinstall` definition:

```
- git:
    local-name: fremen
    uri: https://github.com/strands-project/fremen.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on

continued:



package	maintainer	authors	licence	depends on
fremenserver	Tom Krajnik	Tom Krajnik	MIT	genmsg , catkin , roscpp , actionlib_msgs , message_runtime , actionlib
frongo	Jaime Pulido Fentanes		MIT	fremenserver , catkin , mongodb_store
frongoweb	Marc Hanheide		MIT	catkin , frongo , rospy

bayes_tracking

Source Code: https://github.com/LCAS/bayestracking.git (branch: master)

rosinstall definition:

```
- git:
    local-name: bayes_tracking
    uri: https://github.com/LCAS/bayestracking.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
bayes_tracking	Christian Dondrup, Marc Hanheide	Nicola Bellotto	MIT	catkin , cv_bridge

iliad_smp

rosinstall definition:

```
- git:
    local-name: iliad_smp
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
iliad_smp	Luigi.Palmieri		Proprietary	

people_detection

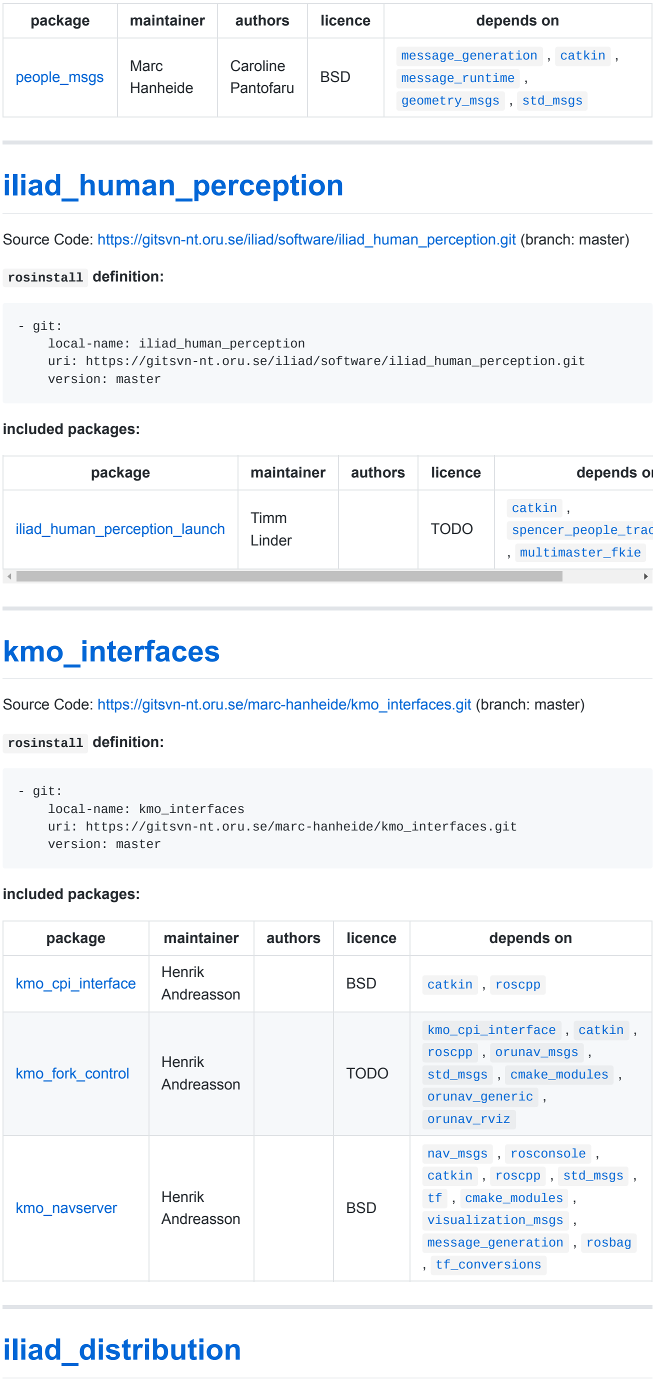
Source Code: https://github.com/lcas/people_detection.git (branch: kinetic-devel)

rosinstall definition:

```
- git:
    local-name: people_detection
    uri: https://github.com/lcas/people_detection.git
    version: kinetic-devel
```

included packages:

continued:



package	maintainer	authors	licence	depends on
people_msgs	Marc Hanheide	Caroline Pantofaru	BSD	message_generation , catkin , message_runtime , geometry_msgs , std_msgs

iliad_human_perception

Source Code: https://gitsvn-nt.oru.se/iliad/software/iliad_human_perception.git (branch: master)

rosinstall definition:

```
- git:
    local-name: iliad_human_perception
    uri: https://gitsvn-nt.oru.se/iliad/software/iliad_human_perception.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends o
iliad_human_perception_launch	Timm Linder		TODO	catkin , spencer_people_trac , multimaster_fkie

kmo_interfaces

Source Code: https://gitsvn-nt.oru.se/marc-hanheide/kmo_interfaces.git (branch: master)

rosinstall definition:

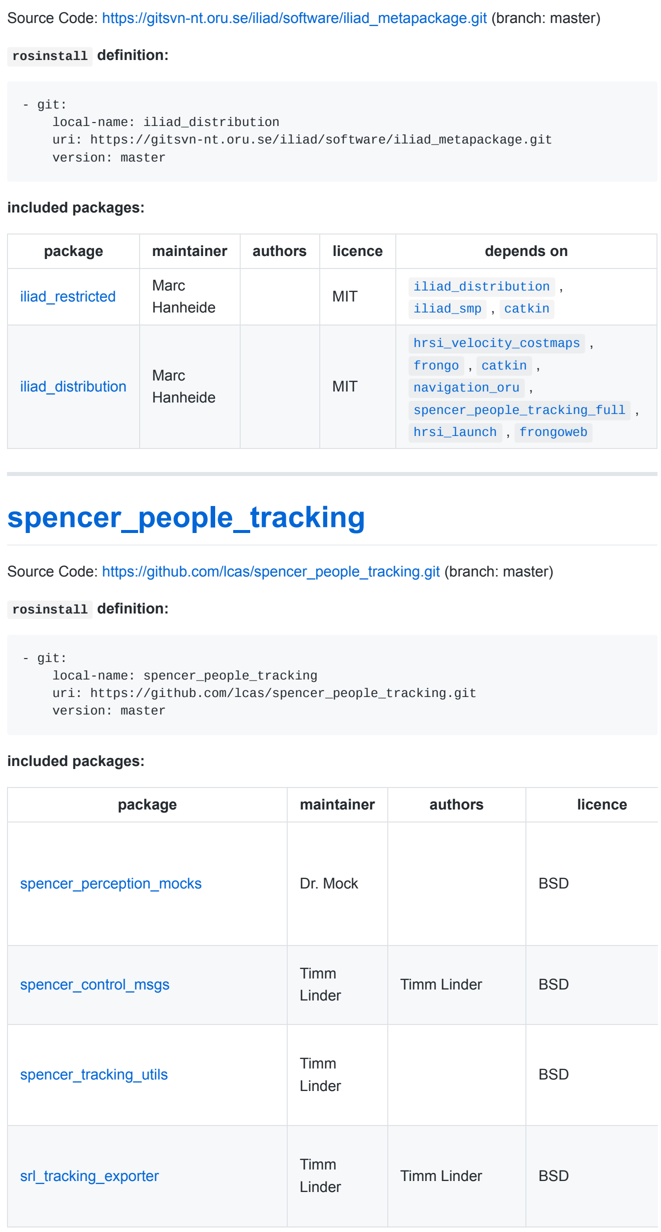
```
- git:
    local-name: kmo_interfaces
    uri: https://gitsvn-nt.oru.se/marc-hanheide/kmo_interfaces.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
kmo_cpi_interface	Henrik Andreasson		BSD	catkin , roscpp
kmo_fork_control	Henrik Andreasson		TODO	kmo_cpi_interface , catkin , roscpp , orunav_msgs , std_msgs , cmake_modules , orunav_generic , orunav_rviz
kmo_navserver	Henrik Andreasson		BSD	nav_msgs , rosconsole , catkin , roscpp , std_msgs , tf , cmake_modules , visualization_msgs , message_generation , rosbag , tf_conversions

iliad_distribution

continued:



Source Code: https://gitsvn-nt.oru.se/iliad/software/iliad_metapackage.git (branch: master)

`rosinstall` definition:

```
- git:
    local-name: iliad_distribution
    uri: https://gitsvn-nt.oru.se/iliad/software/iliad_metapackage.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
iliad_restricted	Marc Hanheide		MIT	`iliad_distribution` , `iliad_smp` , `catkin`
iliad_distribution	Marc Hanheide		MIT	`hrsi_velocity_costmaps` , `frongo` , `catkin` , `navigation_oru` , `spencer_people_tracking_full` , `hrsi_launch` , `frongoweb`

spencer_people_tracking

Source Code: https://github.com/lcas/spencer_people_tracking.git (branch: master)

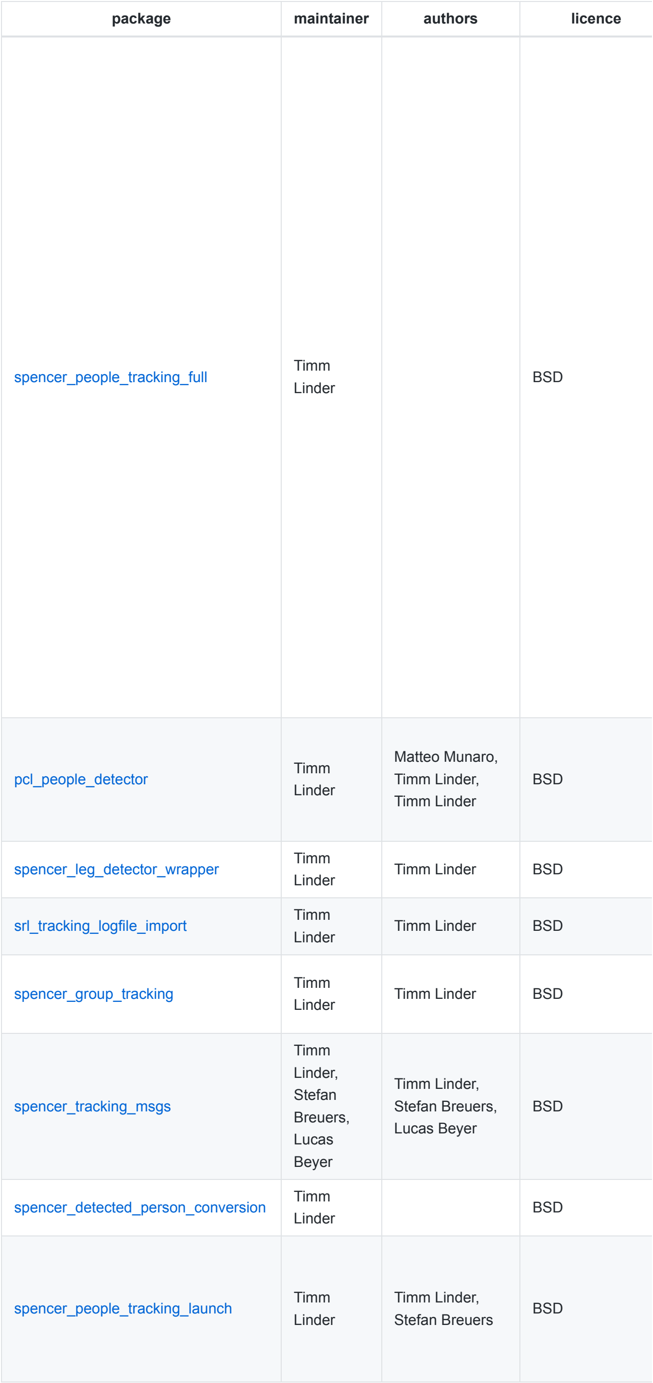
`rosinstall` definition:

```
- git:
    local-name: spencer_people_tracking
    uri: https://github.com/lcas/spencer_people_tracking.git
    version: master
```

included packages:

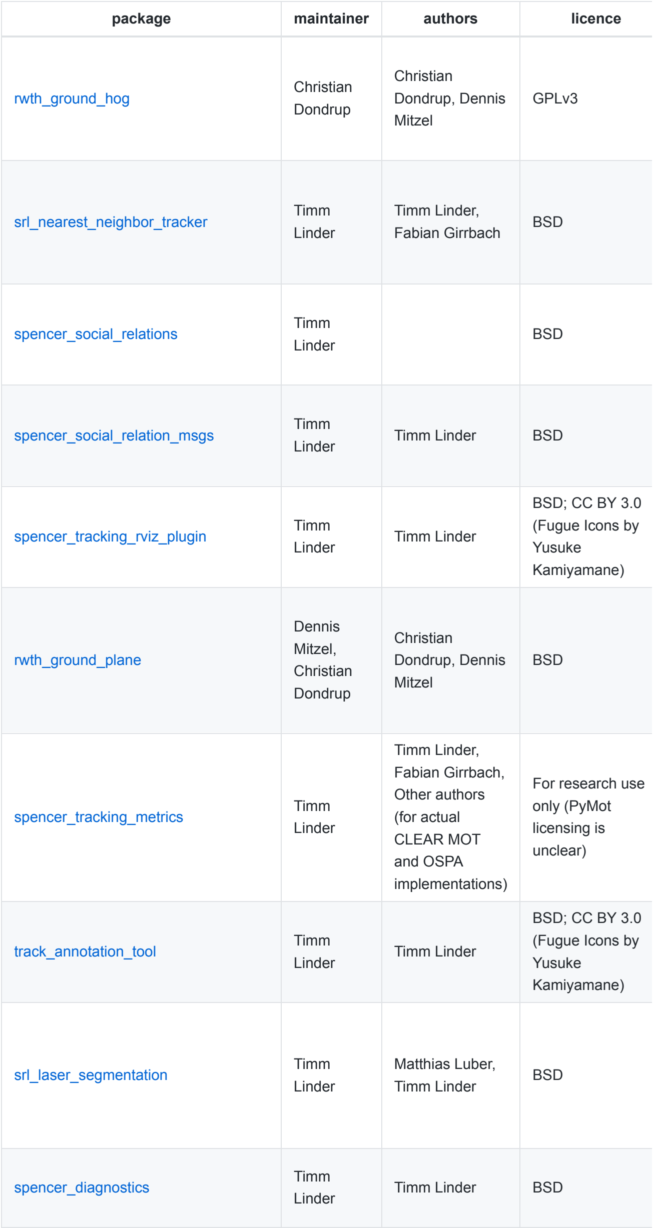
package	maintainer	authors	licence
spencer_perception_mocks	Dr. Mock		BSD
spencer_control_msgs	Timm Linder	Timm Linder	BSD
spencer_tracking_utils	Timm Linder		BSD
srl_tracking_exporter	Timm Linder	Timm Linder	BSD

continued:



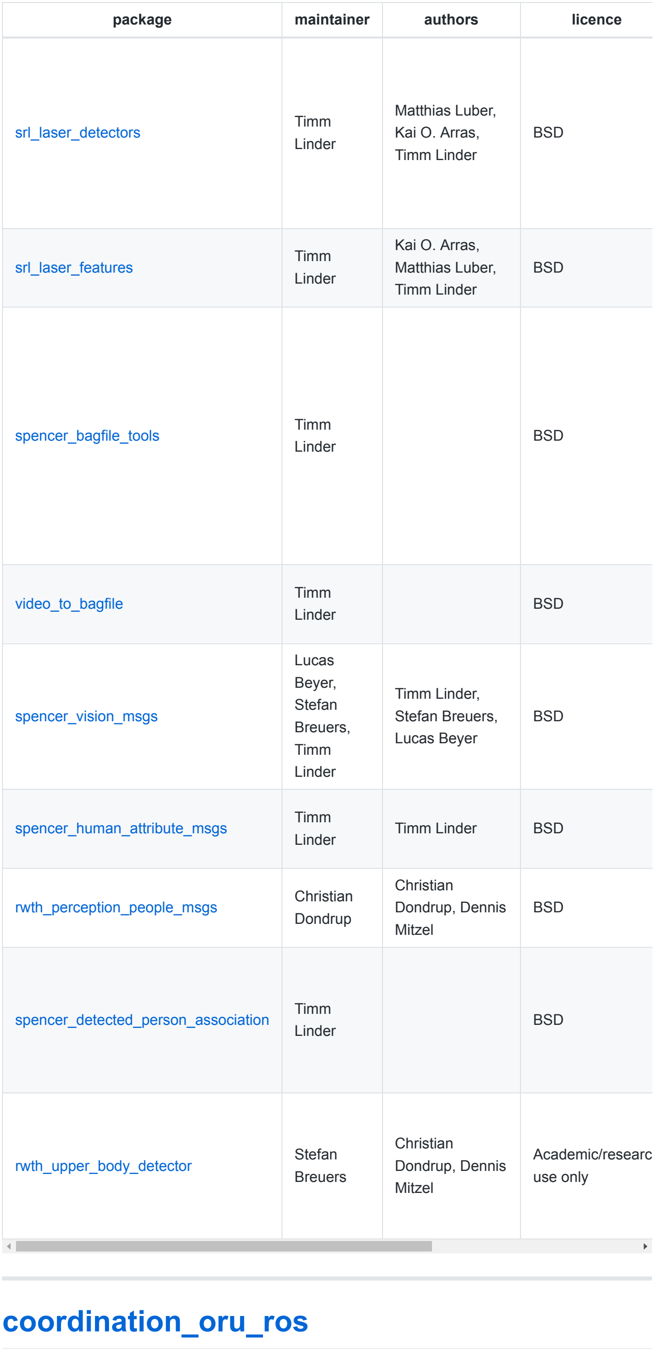
package	maintainer	authors	licence
spencer_people_tracking_full	Timm Linder		BSD
pcl_people_detector	Timm Linder	Matteo Munaro, Timm Linder, Timm Linder	BSD
spencer_leg_detector_wrapper	Timm Linder	Timm Linder	BSD
srl_tracking_logfile_import	Timm Linder	Timm Linder	BSD
spencer_group_tracking	Timm Linder	Timm Linder	BSD
spencer_tracking_msgs	Timm Linder, Stefan Breuers, Lucas Beyer	Timm Linder, Stefan Breuers, Lucas Beyer	BSD
spencer_detected_person_conversion	Timm Linder		BSD
spencer_people_tracking_launch	Timm Linder	Timm Linder, Stefan Breuers	BSD

continued:



package	maintainer	authors	licence
rwth_ground_hog	Christian Dondrup	Christian Dondrup, Dennis Mitzel	GPLv3
srl_nearest_neighbor_tracker	Timm Linder	Timm Linder, Fabian Girrbach	BSD
spencer_social_relations	Timm Linder		BSD
spencer_social_relation_msgs	Timm Linder	Timm Linder	BSD
spencer_tracking_rviz_plugin	Timm Linder	Timm Linder	BSD; CC BY 3.0 (Fugue Icons by Yusuke Kamiyamane)
rwth_ground_plane	Dennis Mitzel, Christian Dondrup	Christian Dondrup, Dennis Mitzel	BSD
spencer_tracking_metrics	Timm Linder	Timm Linder, Fabian Girrbach, Other authors (for actual CLEAR MOT and OSPA implementations)	For research use only (PyMot licensing is unclear)
track_annotation_tool	Timm Linder	Timm Linder	BSD; CC BY 3.0 (Fugue Icons by Yusuke Kamiyamane)
srl_laser_segmentation	Timm Linder	Matthias Luber, Timm Linder	BSD
spencer_diagnostics	Timm Linder	Timm Linder	BSD

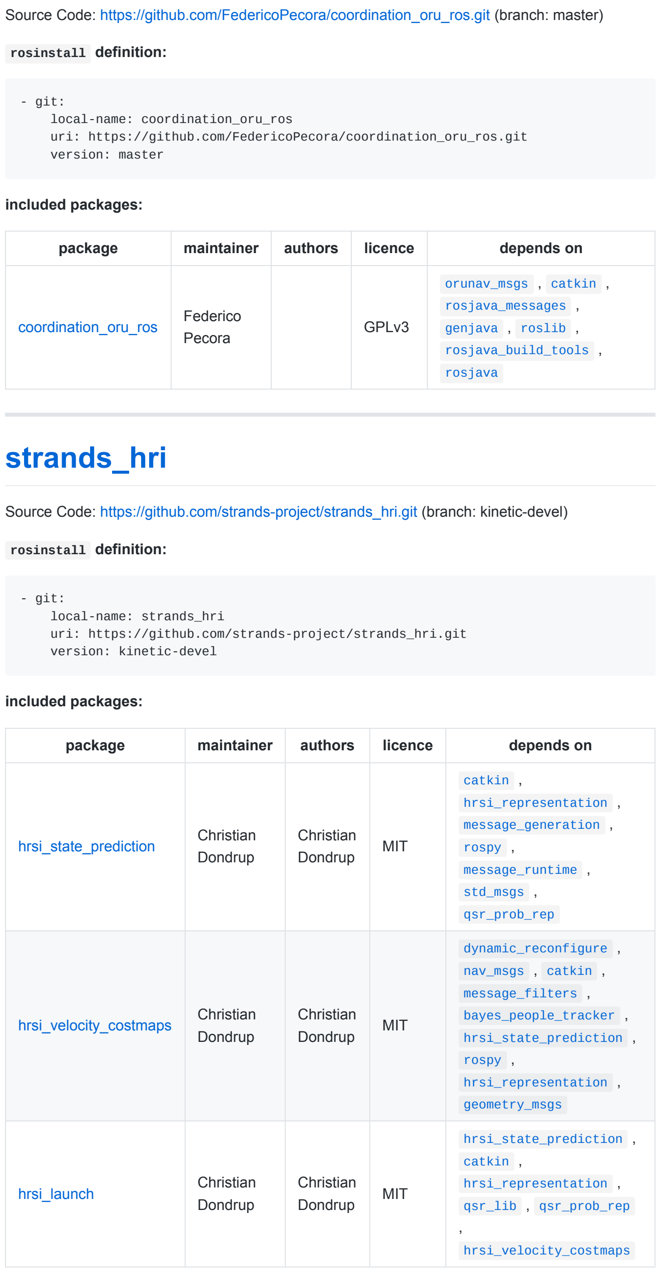
continued:



package	maintainer	authors	licence
srl_laser_detectors	Timm Linder	Matthias Luber, Kai O. Arras, Timm Linder	BSD
srl_laser_features	Timm Linder	Kai O. Arras, Matthias Luber, Timm Linder	BSD
spencer_bagfile_tools	Timm Linder		BSD
video_to_bagfile	Timm Linder		BSD
spencer_vision_msgs	Lucas Beyer, Stefan Breuers, Timm Linder	Timm Linder, Stefan Breuers, Lucas Beyer	BSD
spencer_human_attribute_msgs	Timm Linder	Timm Linder	BSD
rwth_perception_people_msgs	Christian Dondrup	Christian Dondrup, Dennis Mitzel	BSD
spencer_detected_person_association	Timm Linder		BSD
rwth_upper_body_detector	Stefan Breuers	Christian Dondrup, Dennis Mitzel	Academic/researc use only

coordination_oru_ros

continued:



Source Code: https://github.com/FedericoPecora/coordination_oru_ros.git (branch: master)

`rosinstall` definition:

```
- git:
    local-name: coordination_oru_ros
    uri: https://github.com/FedericoPecora/coordination_oru_ros.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
coordination_oru_ros	Federico Pecora		GPLv3	`orunav_msgs` , `catkin` , `rosjava_messages` , `genjava` , `roslib` , `rosjava_build_tools` , `rosjava`

strands_hri

Source Code: https://github.com/strands-project/strands_hri.git (branch: kinetic-devel)

`rosinstall` definition:

```
- git:
    local-name: strands_hri
    uri: https://github.com/strands-project/strands_hri.git
    version: kinetic-devel
```

included packages:

package	maintainer	authors	licence	depends on
hrsi_state_prediction	Christian Dondrup	Christian Dondrup	MIT	`catkin` , `hrsi_representation` , `message_generation` , `rospy` , `message_runtime` , `std_msgs` , `qsr_prob_rep`
hrsi_velocity_costmaps	Christian Dondrup	Christian Dondrup	MIT	`dynamic_reconfigure` , `nav_msgs` , `catkin` , `message_filters` , `bayes_people_tracker` , `hrsi_state_prediction` , `rospy` , `hrsi_representation` , `geometry_msgs`
hrsi_launch	Christian Dondrup	Christian Dondrup	MIT	`hrsi_state_prediction` , `catkin` , `hrsi_representation` , `qsr_lib` , `qsr_prob_rep` , `hrsi_velocity_costmaps`

continued:

package	maintainer	authors	licence	depends on
hrsi_representation	Christian Dondrup	Christian Dondrup	MIT	dynamic_reconfigure , nav_msgs , roslib , catkin , qsr_lib , std_msgs , bayes_people_tracker , sensor_msgs , rostest , geometry_msgs , rosunit , rospy

strands_qsr_lib

Source Code: https://github.com/strands-project/strands_qsr_lib.git (branch: master)

`rosinstall` definition:

```
- git:
    local-name: strands_qsr_lib
    uri: https://github.com/strands-project/strands_qsr_lib.git
    version: master
```

included packages:

package	maintainer	authors	licence	depends on
qsr_lib	Yiannis Gatsoulis, Marc Hanheide		MIT	roslib , catkin , message_runtime , std_msgs , rostest , rospy , rosunit , message_generation
qsr_prob_rep	Marc Hanheide	Christian Dondrup	GPLv2	roslib , catkin , message_runtime , std_msgs , rostest , rospy , rosunit , message_generation , std_srvs

acado

Source Code: https://github.com/LCAS/acado.git (branch: stable)

`rosinstall` definition:

```
- git:
    local-name: acado
    uri: https://github.com/LCAS/acado.git
    version: stable
```

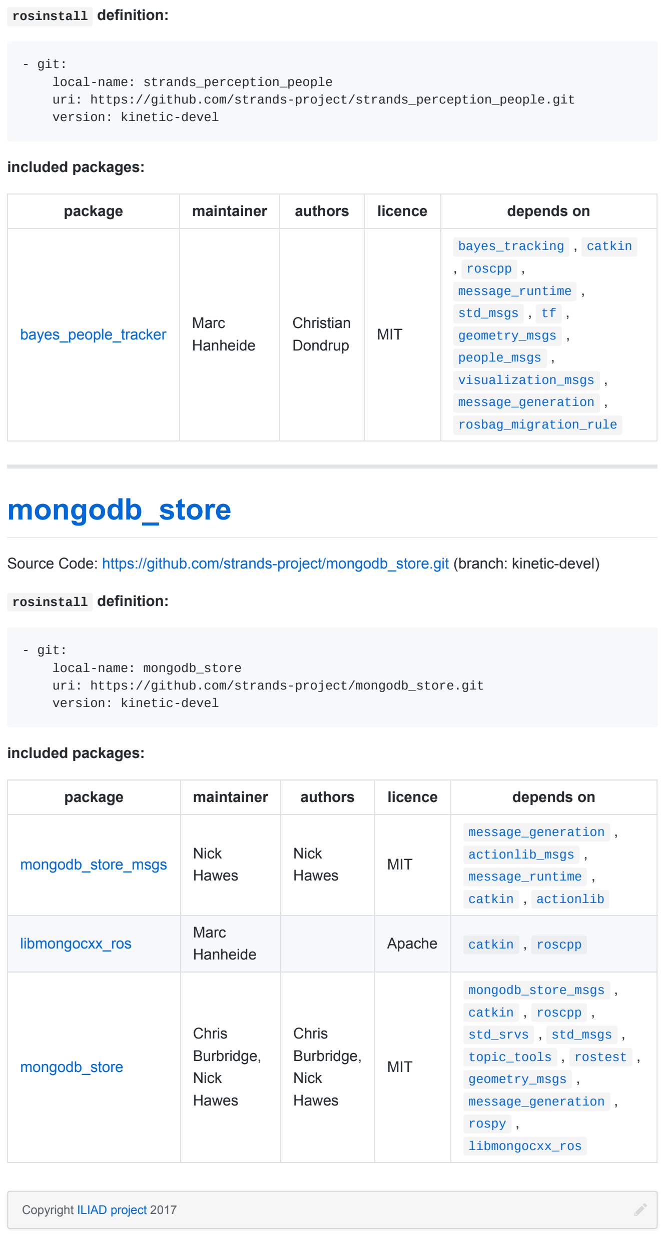
included packages:

package	maintainer	authors	licence	depends on
acado	Marc Hanheide		LGPLv3	catkin

strands_perception_people

Source Code: https://github.com/strands-project/strands_perception_people.git (branch: kinetic-devel)

continued:



`rosinstall` **definition:**

```
- git:
    local-name: strands_perception_people
    uri: https://github.com/strands-project/strands_perception_people.git
    version: kinetic-devel
```

included packages:

package	maintainer	authors	licence	depends on
bayes_people_tracker	Marc Hanheide	Christian Dondrup	MIT	`bayes_tracking` , `catkin` , `roscpp` , `message_runtime` , `std_msgs` , `tf` , `geometry_msgs` , `people_msgs` , `visualization_msgs` , `message_generation` , `rosbag_migration_rule`

mongodb_store

Source Code: https://github.com/strands-project/mongodb_store.git (branch: kinetic-devel)

`rosinstall` **definition:**

```
- git:
    local-name: mongodb_store
    uri: https://github.com/strands-project/mongodb_store.git
    version: kinetic-devel
```

included packages:

package	maintainer	authors	licence	depends on
mongodb_store_msgs	Nick Hawes	Nick Hawes	MIT	`message_generation` , `actionlib_msgs` , `message_runtime` , `catkin` , `actionlib`
libmongocxx_ros	Marc Hanheide		Apache	`catkin` , `roscpp`
mongodb_store	Chris Burbridge, Nick Hawes	Chris Burbridge, Nick Hawes	MIT	`mongodb_store_msgs` , `catkin` , `roscpp` , `std_srvs` , `std_msgs` , `topic_tools` , `rostest` , `geometry_msgs` , `message_generation` , `rospy` , `libmongocxx_ros`

Copyright ILIAD project 2017

References

[1] Georgios Gousios, Martin Pinzger, and Arie van Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 345–355, New York, NY, USA, 2014. ACM.