ISTITUTO
DI TECNOLOGIE DELLA
COMUNICAZIONE,
DELL'INFORMAZIONE
E DELLA
PERCEZIONE

Scuola Superiore
Sant'Anna

# ArchGenTool: A System-Independent Collaborative Tool for Robotic Architecture Design


Emanuele Ruffaldi (SSSA)

I. Kostavelis, D. Giakoumis, D. Tzovaras (CERTH)


CERTH
CENTRE FOR
RESEARCH & TECHNOLOGY
HELLAS

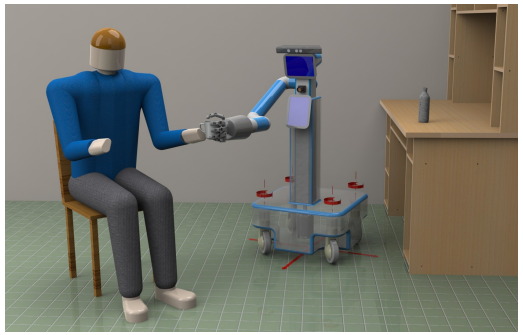
RAMCIP

# Overview

- Problem Statement
- Existing Solutions
- ArchGen Principles
- ArchGen Implementation
- Example
- Conclusions

# Context


- Service Robotics aims at supporting the activity and life of people in home environments
- The H2020 RAMCIP project (2015-2017) targets Mild-Cognitive Impaired people by means of a new Service robot with both manipulative and cognitive capabilities
- Based on ROS the Software Architecture of RAMCIP aims at
  - Understanding the Environment
  - Understanding User State and Action
  - Support the User Activity
  - Controlling Arm with Dexterous Hand
  - Controlling Platform Motion
  - Controlling the Communication Module



- How we can develop and manage such complex system?

# Key Concept

Robotic Architectures are defined along three elements

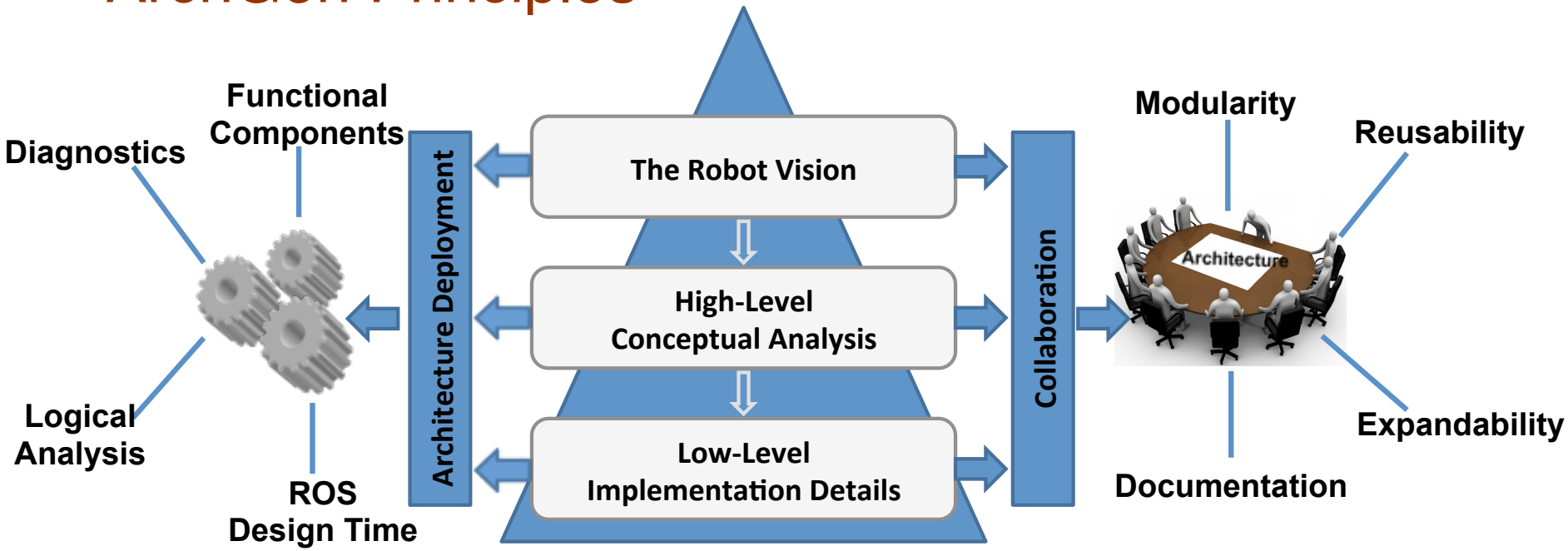
1. **Software components** libraries providing functionalities
2. **Software architecture** that describes the instantiation of the components and their interconnection
3. **Functional Requirements**

# State of the Art

- ROBOTML based on a DSL for robot ontology and integrated with SysML
- BRIDE (BRICS project)
  - Allows to design models of components (ROS/Orocos)
  - Originally purely on Eclipse moving also to textual representation
  - Auto-code generation
- HyperFlex
  - Eclipse based, symbolic representation of architecture points and variants
  - Generation of output to ROS / Orocos
- Robotics Run-time Adaptation Toolchain
  - Run-time Adaptation over HyperFlex
- Architecture Modeling and Analysis Language (AMAL)

Mostly based on Eclipse, with big focus on the specific ROS/Orocos detailed architecture. Wanted a more general tool for refining the architecture

# ArchGen Principles

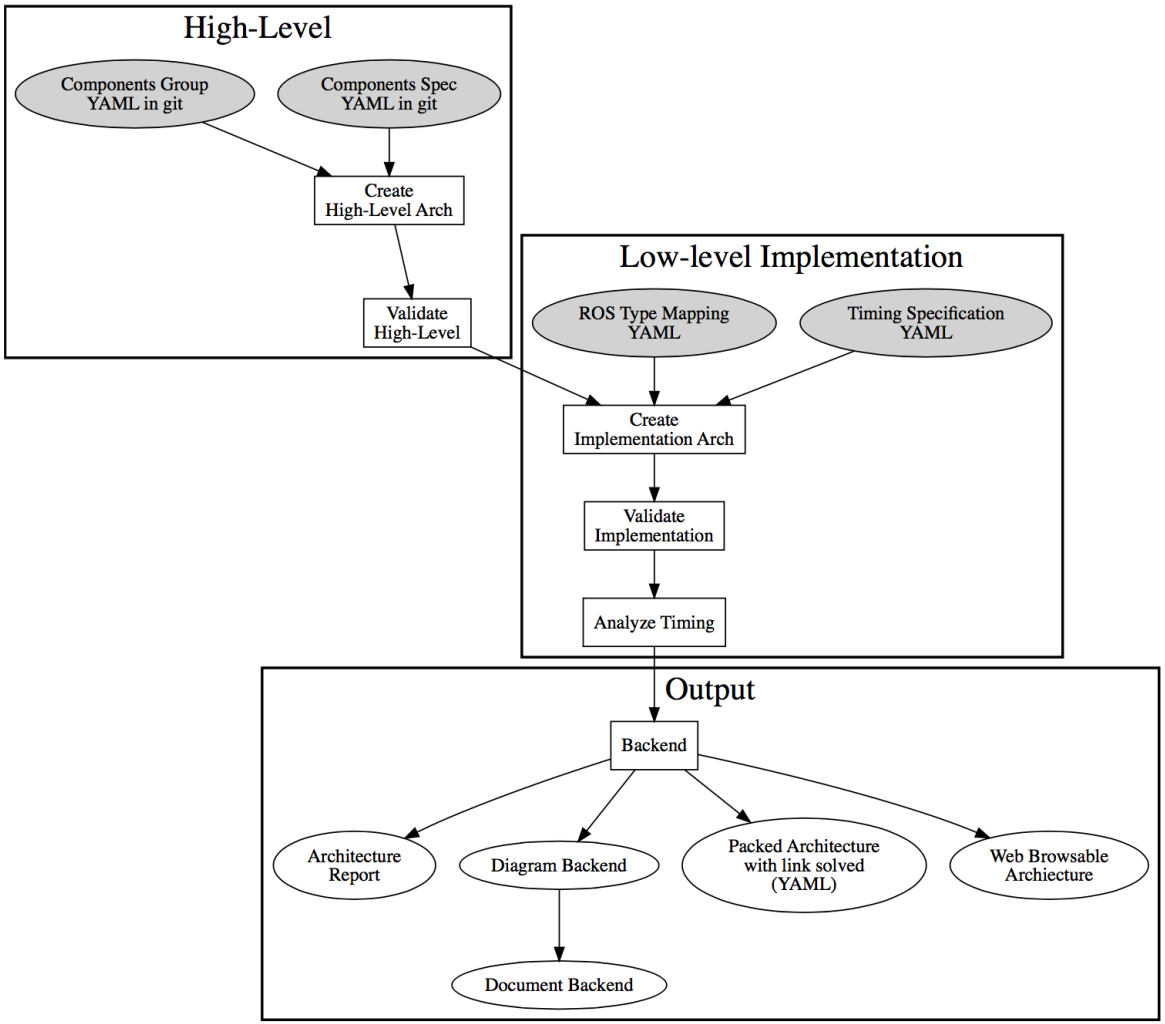



- ArchGen comes as a standalone tool for creating and manipulating Robotic Architecture in a collaborative way
- ArchGen deals with both high-level functional architecture and the low-level implementation (ROS-based)

# ArchGen Features

- Component-based Model
- YAML based representation
- Collaboration via Git (github)
- Multiple back-ends
  - Visualization
  - Report
- Analysis

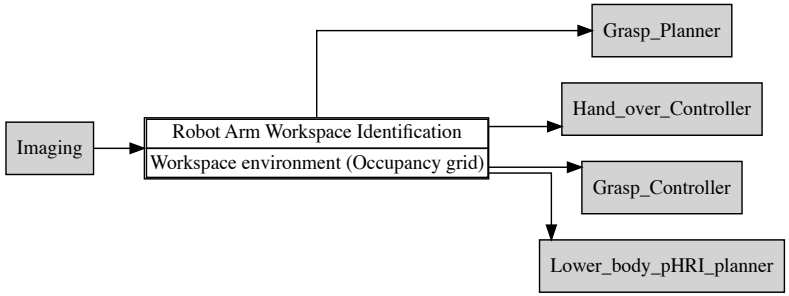
# ArchGen Workflow



# High-Level

- The High-Level functional architecture is the abstract specification of the robot components
  - Component Instances
  - Connected by Data Interfaces
- Additional Properties
  - Performance criteria
  - Operative Requirements

```
- name: Robot Arm Workspace Identification
  group: Current Robot State
  inputs: {Imaging: '*'}
  executionmode: On demand
  functionality: The Robot arm workspace identification functional component is an on-demand process which
                 is responsible to determine the free workspace in the currently observed scene. The component takes
                 into consideration the respective point cloud of the scene and provides an occupancy grid to describe
                 the workspace where the robot arm can move.
  performance:
  - [Accuracy, Depends on the accuracy of the point cloud and the discretization i.e. cell size]
  - [Execution Time, Expected at least 10 frames per second]
  outputs: ['Workspace environment (Occupancy grid)']
  targets:
  oprequirements: It assumes continuous streaming from the Kinect2 sensor placed on-board the RAMCIP platform
    the RAMCIP platform
```



# Tool Output

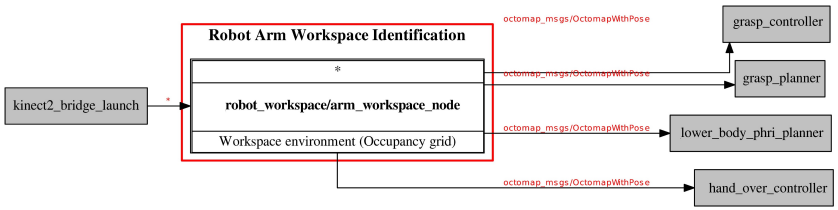
- The High-level execution step produces
  - YAML files for further automatic processing
  - Diagrams for parts
  - Word document
  - Web pages
  - Report of Architecture analysis
- The generated YAML files represent the aggregated Group/All of contributions
- The Word document is used for EU Project reporting

# ROS Mapping

- The implementation phase declares how to realize the functional components, each into one or more ROS components
  - Which existing ROS components are used
  - Their interconnection
  - The mapping of High-level types to ROS types

- Implementation can be put as an attribute of a Functional Component or instead placed in a Separate YAML document
  - Name is the ROS package
  - Execution Method

```yaml
- name: ramcip_robot_workspace/arm_workspace_node
  group: Robot Arm Workspace Identification
  inputs: {kinect2_bridge/kinect2_bridge: '*'}
  executionmode: On demand
  functionality: The Robot arm workspace identification functional component is an on-demand process which
                 is responsible to determine the free workspace in the currently observed scene. The component takes
                 into consideration the respective point cloud of the scene and provides an occupancy grid to describe
                 the workspace where the robot arm can move.
  performance:
  - [Accuracy, Depends on the accuracy of the point cloud and the discretization i.e. cell size]
  - [Execution Time, Expected at least 10 frames per second]
  outputs: ['Workspace environment (Occupancy grid)']
  targets:
  oprequirements: It assumes continuous streaming from the Kinect2 sensor placed on-board the RAMCIP platform
    the RAMCIP platform
```



# Dealing with Time

- Timing is an important aspect of a robotic architecture, in particular for the allocation of hardware resources and the understanding of criticalities
- ArchGen tackles this by specific attributes
  - Mode of timing: Periodic, Event-driven, On-demand (Service)
  - Frequency (Hz)
- Timing of unspecified components is inferred
- ArchGen propagates the timings of signals along the Graph and produces a colored Graph of components similar to the one used in Simulink
  - Message timing
  - Component timing

# Collaboration

- The collaboration is based on the well known code management tools: git/github
  - Each team member writes in common YAML or per-member/partner YAML
    - Architectural changes can be discussed using Issue Management
    - Some properties of the model can be specified in external files (mapping, timings) so that changes can be easily tracked using branches
  - The tool validates the architecture using continuous integration tools (e.g. CircleCI/Shippable)
  - Report and diagrams can be generated online (e.g. Shippable)

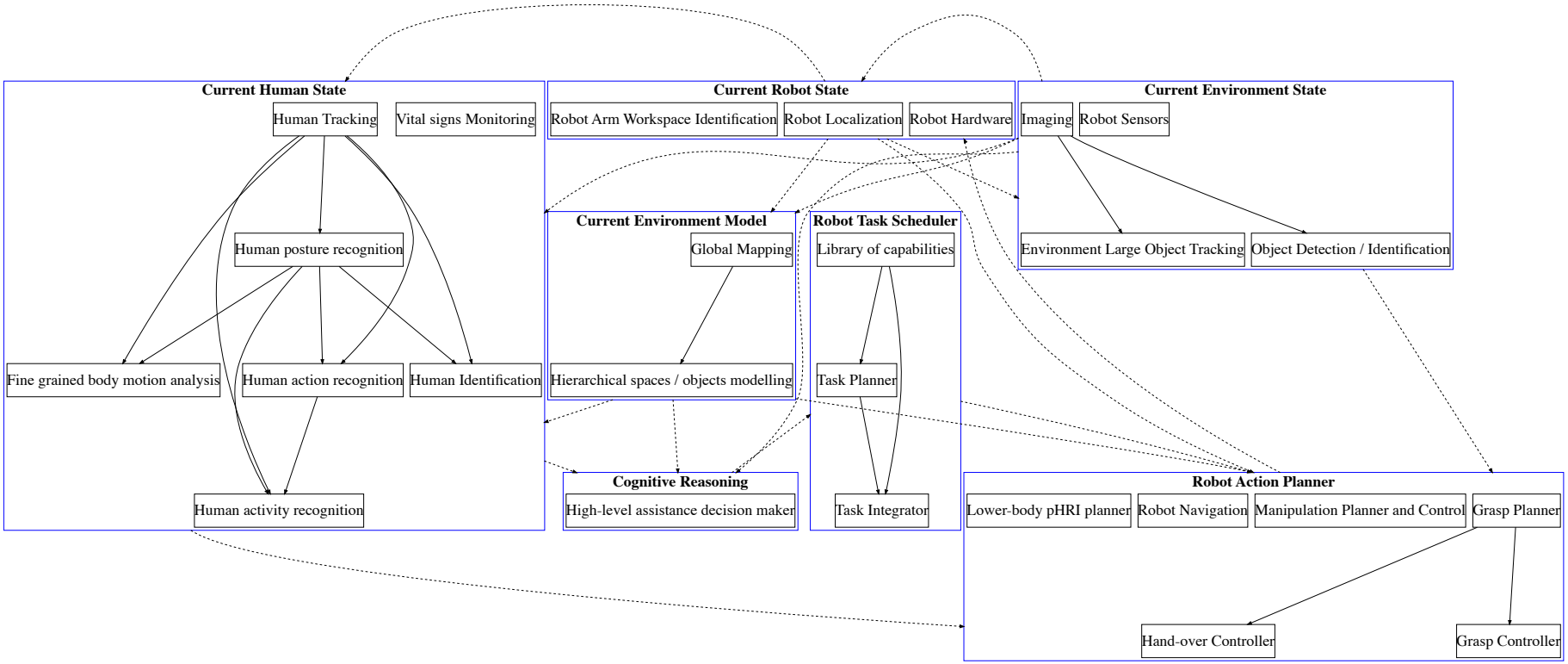
# RAMCIP Case Study

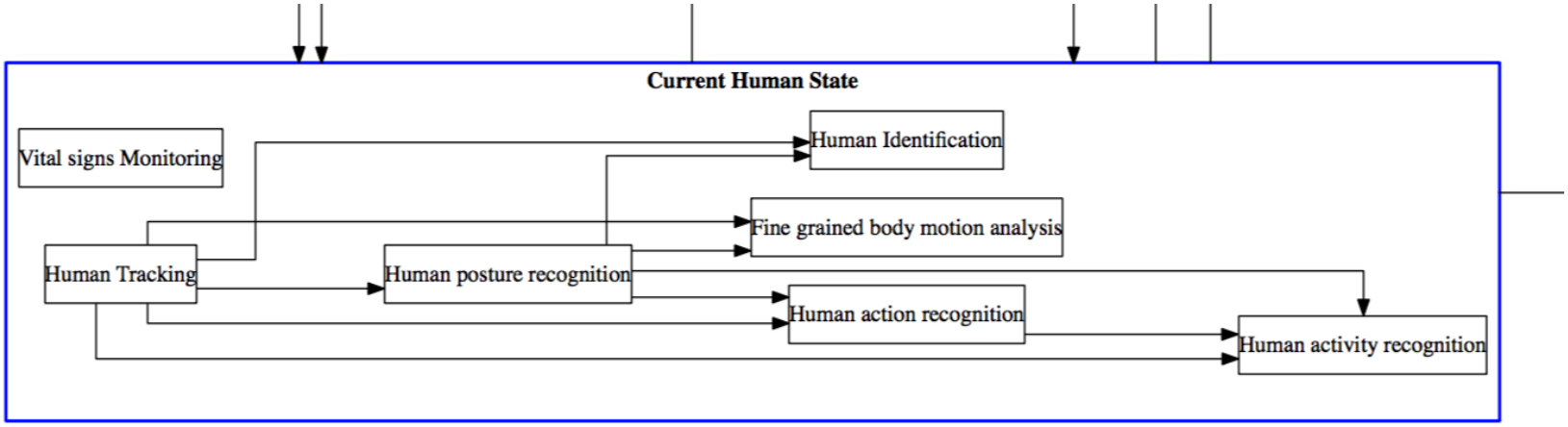
http://www.ramcip-project.eu/

- High-level
  - Component Groups: 10
    - Robot State, Environment State, Environment Model, Human State, Human Model, Cognitive Reasoning, Robot Task Scheduler, Robot Action Planner and Communication Planner
  - Functional Components: 61

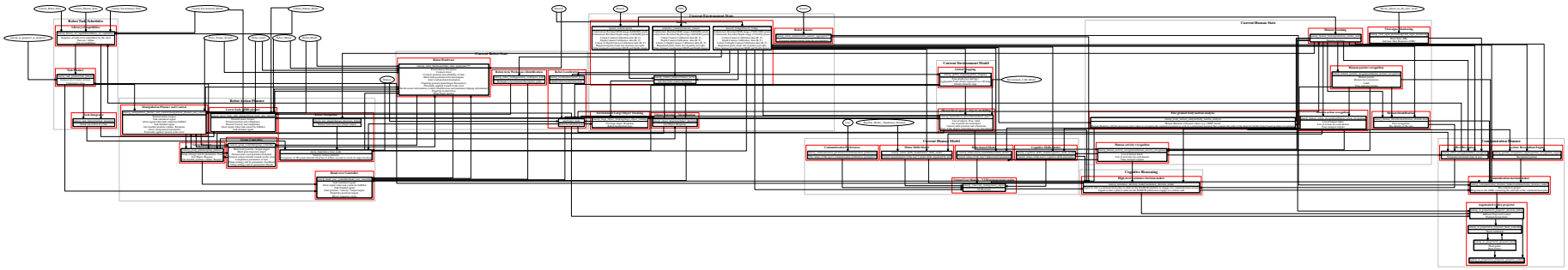
# RAMCIP High-Level Groups

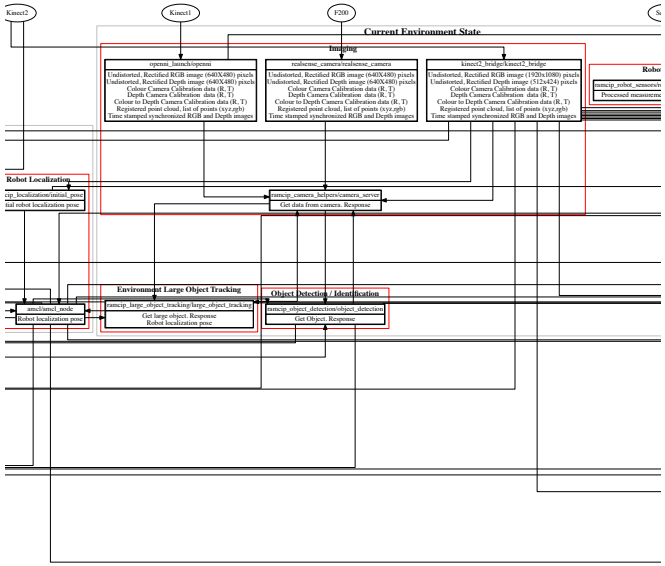
## Except Human Model and Communication





# Overall Diagram





# RAMCIP ROS Mapping Example

Mapping of High-Level to low level types

```
'Undistorted, Rectified RGB image (1920x1080) pixels':
    type: sensor_msgs/Image
    comments: 'A video stream consisting of sensor_msgs/Image messages. These are sta
'Undistorted, Rectified Depth image (512x424) pixels':
    type: sensor_msgs/Image
    comments: 'A video stream consisting of sensor_msgs/Image messages. These are sta
'Colour Camera Calibration data (R, T)':
    type: sensor_msgs/CameraInfo
    comments: 'Ros standard msg.'
'Depth Camera Calibration  data (R, T)':
    type: sensor_msgs/CameraInfo
    comments: 'Ros standard msg.'
'Colour to Depth Camera Calibration data (R, T)':
    type: tf2_msgs/TFMessage
    comments: 'Ros standard msg.'
'Registered point cloud, list of points (xyz,rgb)':
    type: sensor_msgs/PointCloud2
    comments: 'Ros standard msg.'
'Time stamped synchronized RGB and Depth images':
    type: sensor_msgs/Image
    comments: 'A video stream consisting of sensor_msgs/Image messages. These are sta
'Current pose expressed in world coordinates (xyz, rpy)':
    type: geometry_msgs/PoseStamped
    comments: 'Ros standard msg.'
'Time stamped outputs':
    type: 'TBD. Delete this???'
    comments: 'This can be omitted as many ROS msgs contain a Header that is timestam
'The explicit robot model describing its inertials, payload, kinematics':
    type: 'TBD. Probably robot_description URDF in parameter server'
    comments: 'This information is contained in the URDF. Maybe not the payload.'
```

# Conclusion

- ArchGen tries to address the problem of architecture specification in a large robotic problem – while supporting implementation
- Semantics is Component Based with High and Low level
- Testing with an running project

# Future Work

- Making tool Open Source
- Improve Quality of Graphs
- More features
  - Semantic Analysis
  - Verification Tools
  - Integration with ROS
- RAMCIP specific: continuing the project development and integration
- Other projects: REMEDI FP7 is a good candidate

# Thanks! Questions?