

# Mission Control

## Content

Terminology .....	1
Requirements .....	2
Ensemble Formation .....	2
Design Directives / Decisions .....	2
Service Robot mission .....	3
Advertisement .....	3
References .....	3

Bleb bee.

## Terminology

### Components

The design and deployment unit in the ensemble. The component can represent a Robot, a charge station, a sensor, or a NODE in the cloud executing fleet management. A Component is a stateful entity, which accumulates **knowledge** about the state of the world.

### Knowledge

A set of fields that represent state of the, component or external world relevant to the operation of the component.

### Processes

Are the computation realized by the components. Processes are stateless ROS nodes that receive as input fields of the knowledge and write fields back to the knowledge. A definition of a process has

- a map of the input fields
- a map of output fields
- a scheduling scheme (i.e., when / with how frequency the process will be triggered)

### Ensemble

A group of components with a common goal. An Ensemble has a coordinator and members. The membership of a candidate component is decided by a membership function that operates on the state of the ensemble and the candidate component. An ensemble is formed dynamically and opportunistically (rather than statically by design) based on the state of the components.

### Ensemble Formation

How ensembles are formed

- Open Any component matching criteria will be invited to enter the ensemble.

- Utility Based The best known available component according to a utility function.

### Trigger

what motivates the formation of the ensemble

- When components are discovered
- In response to a request

## Requirements

### Ensemble Formation

- Components must advertise its state so another component "hosting" and ensemble can decide on "inviting it to the ensemble"
- A coordinator must listen to advertisements

===

## Design Directives / Decisions

In priority order:

### Don't be redundant with ROS2 functionality

I.e., when possible use the abstractions already provided by ROS (e.g., process scheduling can use node machinery)

### Stateful Component

ros\_ensemble is a stateful component so other components do not need to.

### Encapsulate low-level details about communication (e.g., protocols, used port, dds distribution)

ros\_ensemble is a component that provides an ensemble collaboration mechanism on top of ROS, abstracting other nodes of multi-robot communication.

### Avoid being locked to a specific ROS2 version or DDS provider

Use standard ROS mechanism when possible, to avoid lockin to underlying mechanisms.

### Customizable

the end-user must provide their own knowledge model, and algorithms to map knowledge to the input of other nodes, so this node must be customizable at the code level (e.g., a config file is not sufficient).

### Single thread

Every internal process is a non-blocking process and the system operates in a single thread. This should free ros\_ensemble of the overhead of scheduling threads and avoid problems with concurrent access to the knowledge. === Scenarios

## **Service Robot mission**

Coordinator deployed in a central node that creates ensembles to execute missions on behalf of users.

## **Advertisement**

ComponentA advertise its availability for its initialized roles, which is received by

## **References**