Active Resources on the CPS Virtual Organization

2016 CPS PRINCIPAL INVESTIGATORS' MEETING

Chris.vanBuskirk@vanderbilt.edu

31-Oct-2016

Tel (615) 343-7472 Fax (615) 343-7440
1025 16th Avenue South|Nashville, TN 37212
www.isis.vanderbilt.edu

VANDERBILT UNIVERSITY

THE VO PHASE 2

While the VO portal in its initial embodiment is a useful resources that promotes collaboration and dissemination of information,

The central objective of the proposed research is to transform the CPS-VO from a collaboration platform and passive repository of information into an active resource that provides access to tools and methods emerging from the CPS research community.

The project will make a significant contribution to education via support to student competitions that will help prepare a new generation of students who will be inspired and trained to realize the promise of CPS. We expect that the integrated suite of models, integration platforms, and intellectual frameworks to be developed and contributed by the research community will lead to a new era of low-cost, distributed and open design infrastructure.

An *Active Resource*

- A web-accessible, domain-specific tool-suite that at runtime is executed within a single compute node. These tools, as deployed, are often highly constrained so as to enable end users to concentrate on a very specific problem, without being overwhelmed by extraneous details (i.e. requires *scaffolding* beforehand to reduce complexity).

- Compute Node(s) Are Either:
  1. One or more long-lived, dedicated, shared resources that live somewhere in the world (with simple round-robin scheduling)
  2. A set of servers within some cloud based computing platform (possibly hosted by existing VO infrastructure)
  3. Or, for more complex situations that require load balancing based upon unpredictable demand, an ephemeral, dynamically allocated *application as a service* container within the VO's private cloud maintained at Vanderbilt University

EXAMPLES OF ACTIVE RESOURCES

#1: Reproducible Results

Software Sustainability Institute

About | Blog | Community | Policy | Software | Training | Resour

The Recomputation Manifesto

By Ian Gent, Professor of Computer Science, University of St Andrews.

At the start of this year there was a wonderful stream of tweets with the hashtag #overlyhonestmethods. Many scientists posted the kind of methods descriptions which are true, but would never appear in a paper. My favourite is this one from Ian Holmes.

Although every scientific primer says that replication of scientific experiments is key, to quote this tweet, you'll need luck if you wish to replicate experiments in computational science. There has been significant pressure for scientists to make their code open, but this is not enough. Even if I hired the only postdoc who can get the code to work, she might have forgotten the exact details of how an experiment was run. Or she might not know about a critical dependency on an obsolete version of a library.

The current state of experimental reproducibility in computer science is lamentable. The result is inevitable: experimental results enter the literature which are just wrong. I don't mean that the results don't generalise. I mean that an algorithm which was claimed to do something just does not do that thing: for example, if the original implementation was bugged and was in fact a different algorithm. I suspect this problem is common, and I know for certain that it has happened. Here's an example from my own research area, discovered by my friend and tenacious pursuer of replication Patrick Prosser.

How it should be: the Recomputation Manifesto

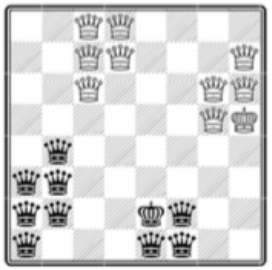
Handbook of Constraint Programming
Edited by F. Rossi, P. van Beek and T. Walsh

Chapter 10

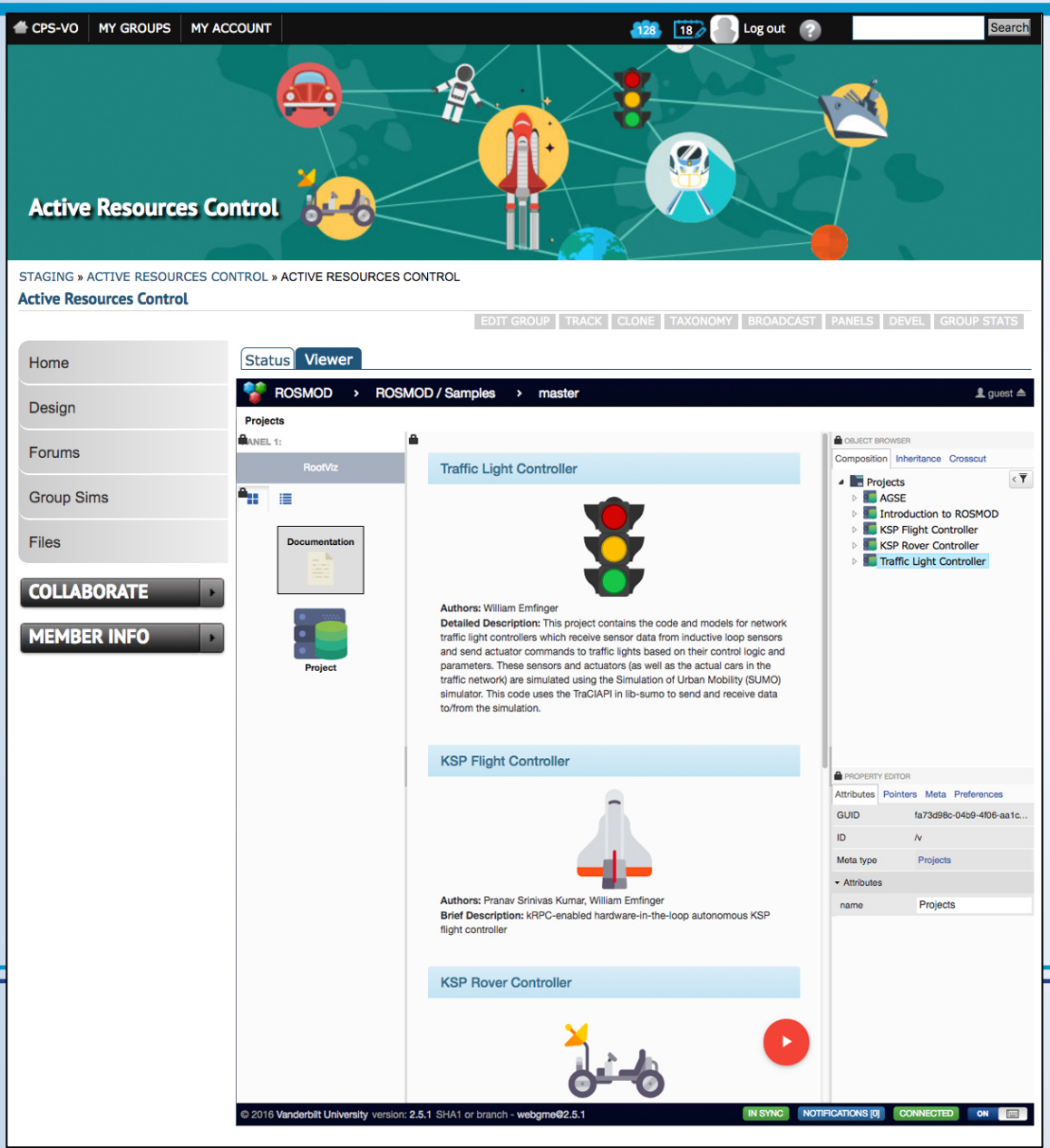
Symmetry in Constraint Programming

Ian P. Gent, Karen E. Petrie, Jean-François Puget

Symmetry in constraints has always been important but in recent years has become a major research area in its own right. A key problem in constraint programming has long been recognised: search can revisit equivalent states over and over again. In principle this problem has been solved, with a number of different techniques. As we write, research remains very active for two reasons. First, there are many difficulties in the practical application of the techniques that are known for symmetry exclusion, and overcoming these remain important research problems. Second, the successes achieved in the area so far have encouraged researchers to find new ways to exploit symmetry. In this chapter we cover both these issues, and the details of the symmetry exclusion methods that have been conceived.



#2: Design/Testing of ROS-based Systems



An active resource that promotes the use of graphical models with well defined semantics for the design and testing of CPS systems based on the ROS middleware (i.e. all code is synthesized from the formal models).

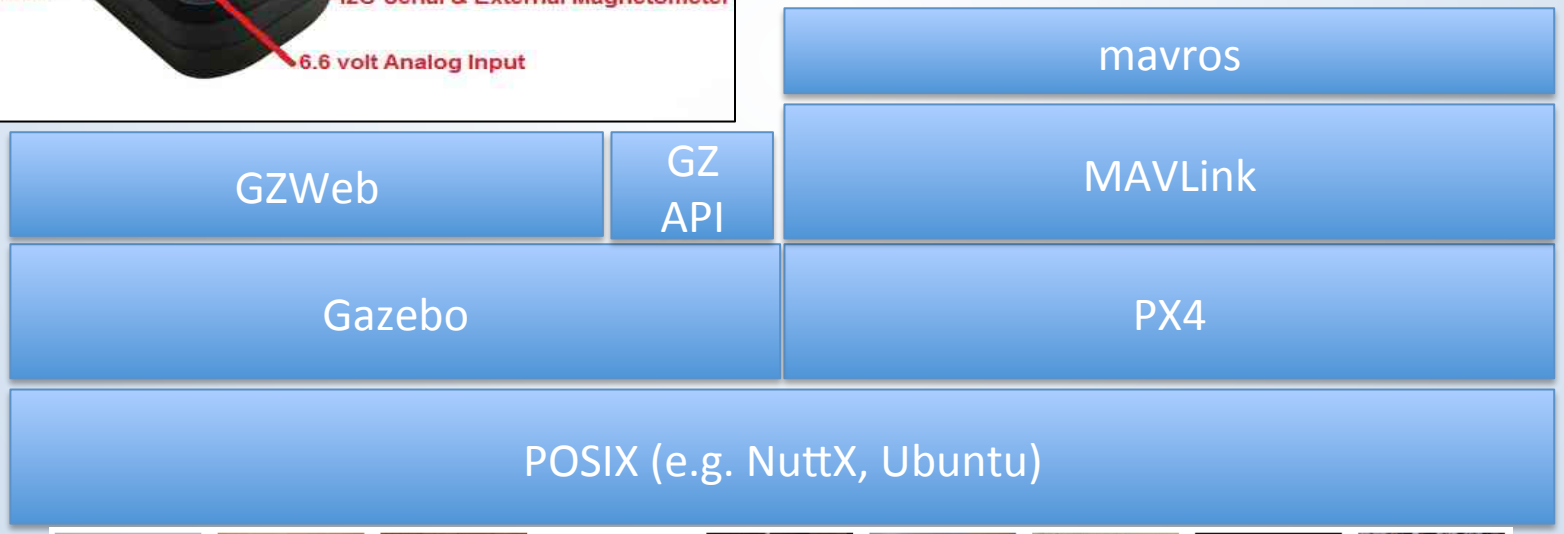
The example application shown here is analysis of various types of cyber attacks on specific configurations of automotive traffic networks.

VANDERBILT UNIVERSITY

#3: The 2016 Student Competition

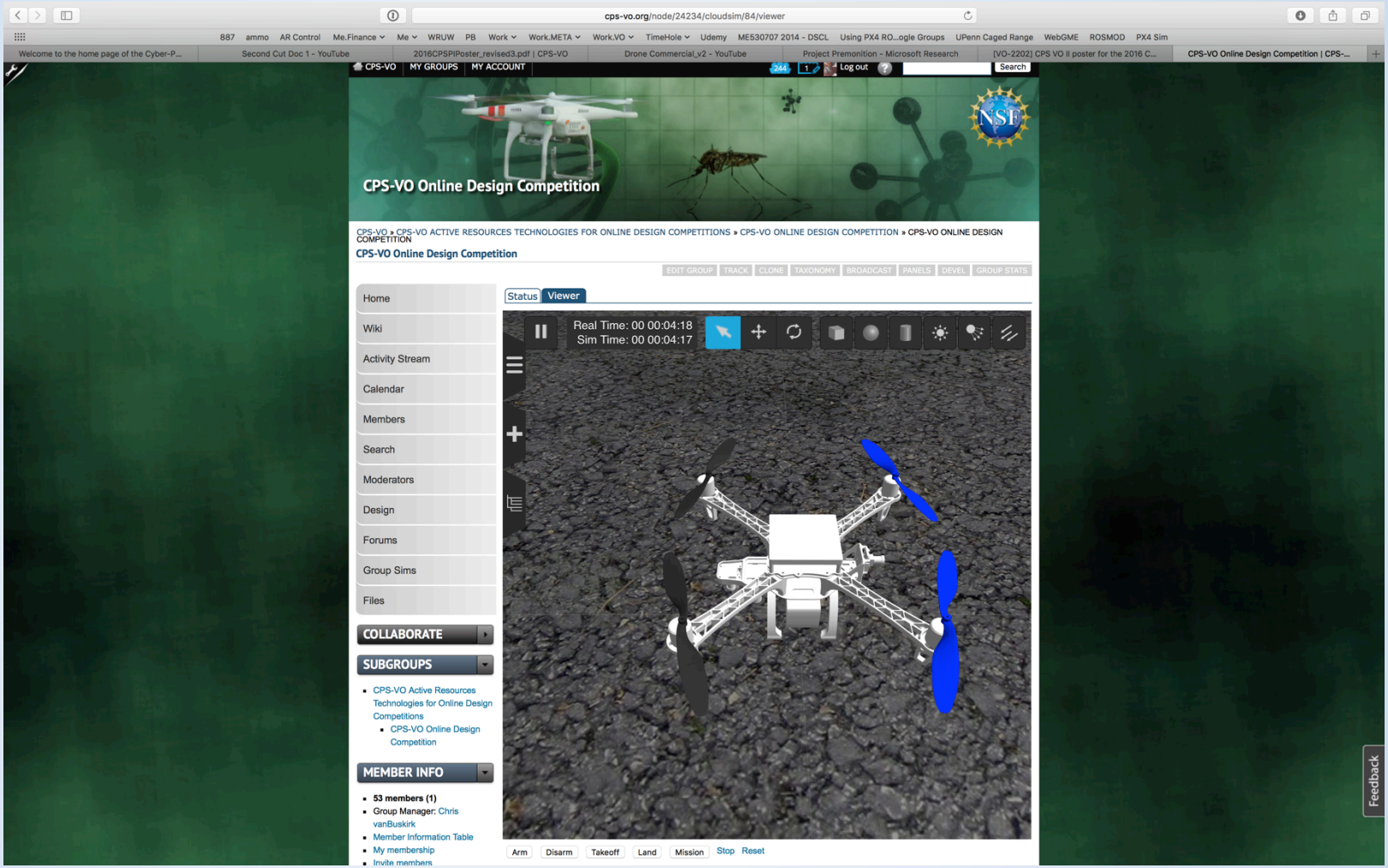
Simulating *Pixhawk* Devices

(open-hardware autopilot)

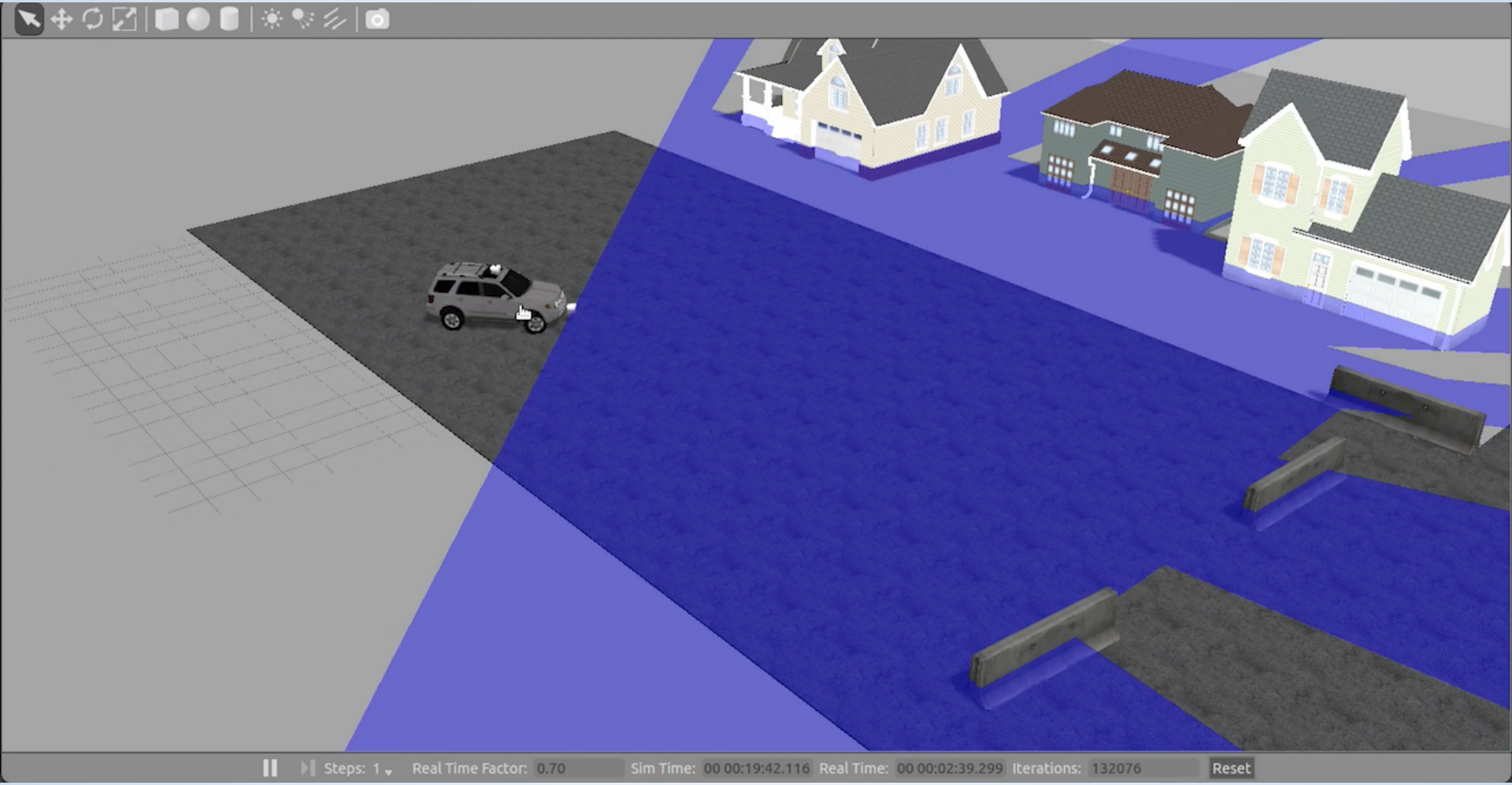


#3: The 2016 Student Competition

PX4 Simulator



#4: The 2017 Spring Competition



Patterns & Templates

DEPLOYING ACTIVE RESOURCES

Deployment: Tool Providers Submit

1. A 'readme' file describing the archetype experiment (incl. licensing information)
2. A declarative specification for provisioning computational resources
3. A declarative specification of the free parameters within a single experiment
4. An archetype set of #!shebang scripts for controlling the lifecycle of an experiment

Going Forward

- http://cps-vo.org/group/2017CATVehicleChallenge

Interested in competitions in general? Suggestions for new competitions?

- competitions@cps-vo.org

Ideas for submitting a new tool to be incorporated as an Active Resource on the VO?

- active-resources@cps-vo.org

** 2017 Drone Challenge to be announced…*

THANK YOU

USING ACTIVE RESOURCES WITHIN VO GROUPS

Previous 'Group Templates' Feature

- On the VO, we already maintain a library of templates for commonly employed use cases applied to groups
  - Workshops: typically require a registration form
  - Research Projects: typically use wikis
- We can instantiate new groups by cloning these templates with the push of a button

Consider A *Challenge Problem* Scenario

1. Similarly, the organizers build up a group that
   - Lays out the rules of the game in the group wiki
   - Provides a discussion forum for Q&A
   - Is seeded with input datasets for various challenges
   - Wires up one or more ActiveResources that provide tool support for the challenges

2. As new teams enter the competition, the above template can be used to quickly spin up a new group, which is a private working space for the individual team.

Provisioning

ansible — view install-px4.yml — 85×40

roscore http://localho... | roscore — java -Xms2... | ...-darwin-x64 — -bash | ...cypher/trunk — -bash | view install-px4.yml

```
##############################################################################
## Gazebo Binaries (incl. gazebo_ros)
##############################################################################
  - name: "Register the OSR Foundation package repo"
    become: yes
    apt_repository: repo="deb http://packages.osrfoundation.org/gazebo/ubuntu-stable
{{ lsb_release }} main" update_cache=yes
    tags: gazebo

  - name: Setup Gazebo package key
    become: yes
    shell: wget http://packages.osrfoundation.org/gazebo.key -O - | apt-key add -
    tags: gazebo

  - name: Install Gazebo version 6
    become: yes
    apt: name=gazebo6 force=yes
    tags: gazebo

  - name: Install dependencies for developing Gazebo code
    become: yes
    apt: name=libgazebo6-dev force=yes
    tags: gazebo

  - name: Install Gazebo_ROS
    become: yes
    apt: name=ros-indigo-gazebo6-ros-pkgs state=latest force=yes
    tags: gazebo


##############################################################################
## MAVROS from source
##############################################################################
  - name: MAVROS dependency - python-wstool
    become: yes
    apt: force=true name=python-wstool
    tags: mavros
  - name: MAVROS dependency - python-rosinstall-generator
    become: yes
```

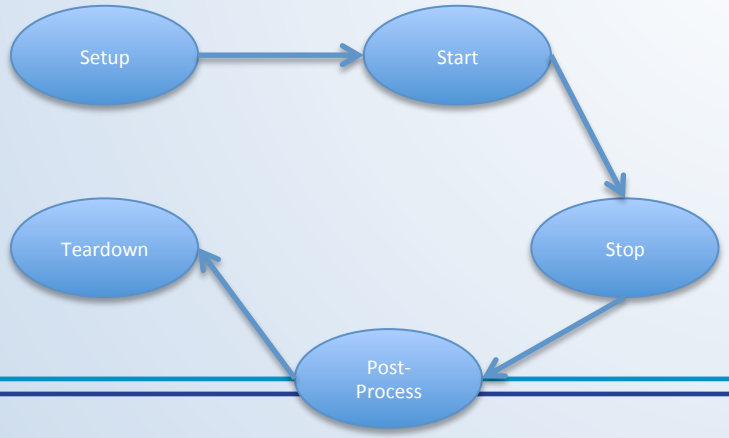
Experiment Parameterization

ansible — vi inputs.yml — 104×34

roscore http://loc... | roscore — java -X... | ...rwin-x64 — -bash | ...125_NSF — -bash | view inst...-px4.yml | vi inputs.yml | ...oy/ansible — -bash

```
---
- tool:  PX4 Simulator
  shortname: PX4
  visibility: private
  - group: 22809  # group/vortex
  - group: 24234  # group/vortex/competition


  inputs:
  - name: HEADLESS_MODE
      type: boolean
      default: true

  - name: WORLD_FILE
    label: World
    type: File.VO
    default: http://cps-vo.org/file/26062/download/78331

  - name: UAV
    label: Vehicle
    type: File.URL
    default: https://raw.githubusercontent.com/cpsvo/sitl_gazebo/master/models/f450/f450.sdf

  - name: TRAP-
    label: Other Model(s)
    multiplicity: 0..10
    type: File.VO
    default: http://cps-vo.org/file/26063/download/79037
    attribute: INITIAL_POS
      - type: triple

  - name: MAX_SIM_DURATION
    type: Integer
:1
```

Filesystem Structure for an Execution Instance of an Active Resource



- +experimental-parameters.yml
- +simulator.yml
- bin
  - postprocess.sh
  - setup.sh
  - start.sh
  - stop.sh
  - teardown.sh
- inputs
  - empty_world.launch
  - F450.sdf
- monitors
  - detect-high-vibration
  - detect-mission-completed
  - watchdog-simulation-snafu
- outputs
  - gazebo.log
  - ros.bag
- state
  - pids
    - 27201
    - 27203
    - 27204
    - 27207
  - TEARING_DOWN