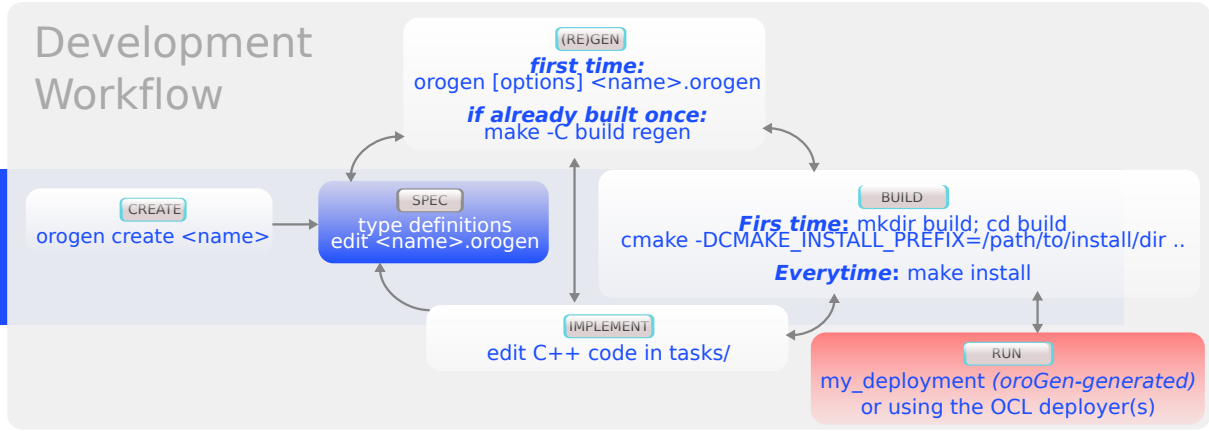
oroGen Cheat Sheet

oroGen v2.x / sheet v1.0

Development Workflow



Main Scope

Typography:
description, oroGen specification, C++ code

Project Information

General oroGen project information
```
name "project_name"
version "0.1"
```

Types

oroGen can use C++ types ...

From a C++ library that exports a pkg-config file
```
using_library "pkg_name"
import_types_from "header_file.h"
```

From a local C++ header
```
import_types_from "header_file.h"
```

From another oroGen project
```
import_types_from "project_name"
```

Tasks

```
task_context "ClassName" do
  # task definition statements
end
```

Defines a subclass of RTT::TaskContext
project_name::ClassName
It is defined in
tasks/ClassName.hpp and
tasks/ClassName.cpp

Deployments

```
deployment "name" do
  # deployment statements
end
```

Generates a corresponding binary which deploys the specified tasks

Task Definitions

Properties

```
property('name', 'type::Name')
property('name', 'type::Name', def_value)

// Read the property
type::Name sample = _name.get();
// Write the property
_name.set(sample);
```

Input Ports

```
input_port('name', 'type::Name')

type::Name sample;
if (_name.read(sample) != RTT::NoData)
{
    // there either a new, or an already-read
    // sample on _name.
}
if (_name.read(sample) == RTT::NewData)
{
    // there was a never-read sample
    // on _name
}
if (_name.connected())
{
    // do something only if the port
    // is connected
}
```

Output Ports

```
output_port('name', 'type::Name')

type::Name sample;
// write data into 'sample'
_name.write(sample);


if (_name.connected())
{
    // do something only if the port
    // is connected
}
```

State Machine

```
needs_configuration
```
The task context starts in PRE_OPERATIONAL, i.e. configureHook() has to be called

Sub-states of RUNNING
```
runtime_states 'diving', 'searching'
runtime(diving)
```
Sub-states of RUNTIME_ERROR
```
error_states 'heat_throttling', 'self_test'
error(heat_throttling)
```
Sub-states of EXCEPTION
```
exception_states 'io_error', 'com_error'
exception(com_error)
```
Sub-states of FATAL_ERROR
```
fatal_states 'internal_error'
fatal(internal_error)
```

Triggering
(works hand-in-hand with deployments)

Port-driven tasks
This can be combined with fd_driven and triggered activities, but won't work on periodic activities
```
port_driven
```
Task will be triggered when data arrives on all input ports declared before the statement
```
port_driven 'port_name'[, 'port_name']
```
Task will be triggered when data arrrives on the specified input ports

Default and required activites
'policy_type' can be either 'triggered', 'fd_driven' or 'periodic'. In the case of 'periodic', policy_options is the period in seconds
```
default_activity policy_type[, policy_options]
```
This triggering mechanism will be used if none is specified in the deployment.
```
required_activity policy_type[, policy_options]
```
This triggering mechanism has to be used.

Deployments

```
task = task('name', 'project::Task')
```
Adds a new task instance of type project::Task using its default activity type

```
task.periodic(0.1)
task.fd_driven
task.triggered
```
Overrides the activity of a deployed task

```
task.realtime
```
Places a deployed task in the OS realtime domain

```
task.priority(value)
task.highest_priority
```
Sets the priority of the deployed task. **value** is an integer between 0 and 99. It is only valid for realtime tasks.

IO-Driven tasks
```
fd_driven
```
Task will be triggered when data arrives on some file descriptors.

Must be set up (usually in configureHook) with
```
RTT::extras::FileDescriptorActivity* fd_activity =
    getActivity<RTT::extras::FileDescriptorActivity>();
fd_activity->watch(fd1);
fd_activity->watch(fd2);
fd_activity->watch(fd3);
fd_activity->setTimeout(value_in_ms);
```
Then, in updateHook()
```
RTT::extras::FileDescriptorActivity* fd_activity =
    getActivity<RTT::extras::FileDescriptorActivity>();
fd_activity->hasTimeout() // one FD has timeout
fd_activity->hasError() // one FD has error
fd_activity->isUpdated(fd1) // fd1 has data on it
```