OML Overview

Max Ott

NICTA

Australian Government

Department of Communications, Information Technology and the Arts

Australian Research Council

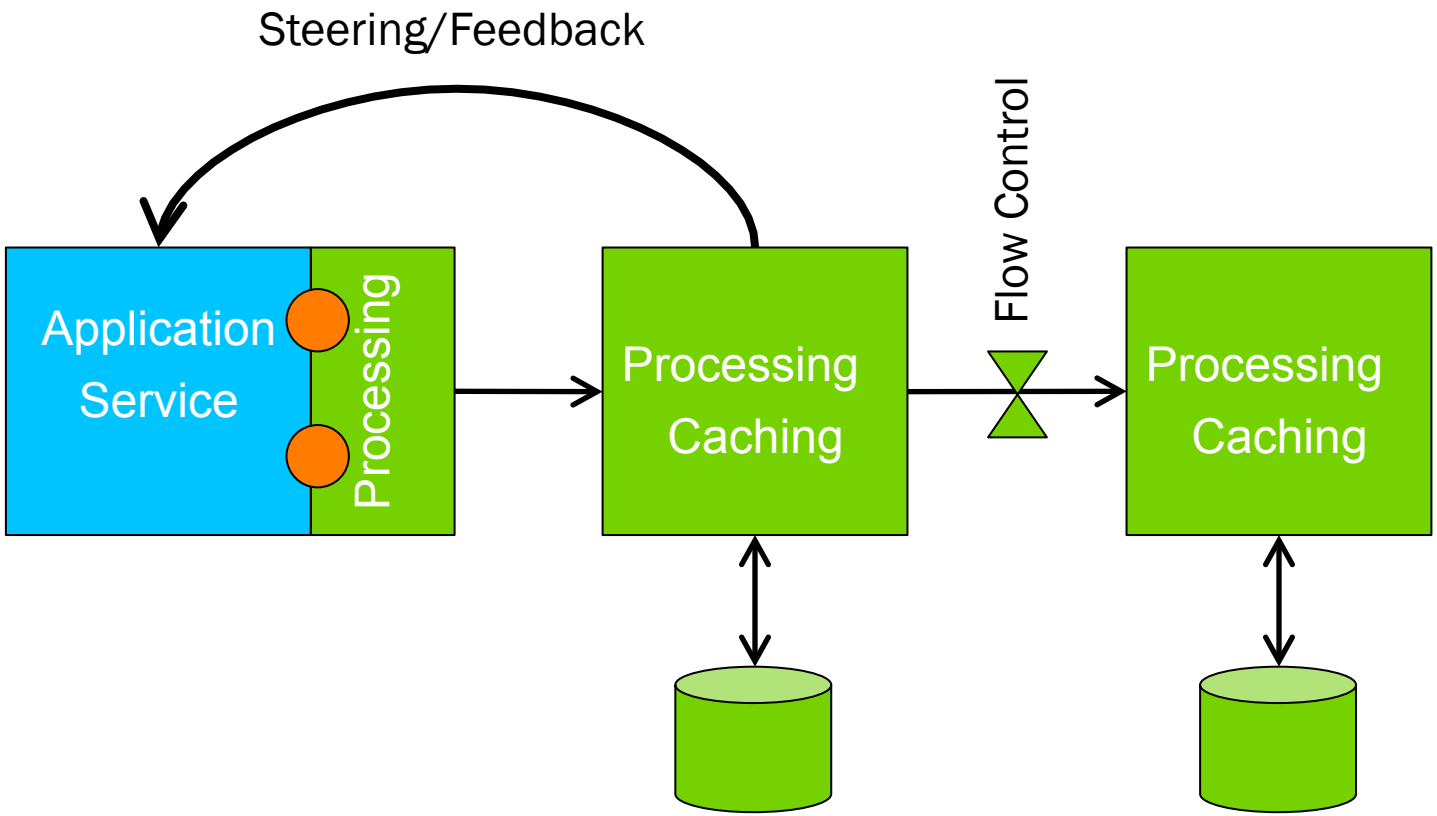
NICTA Members

NICTA Partners

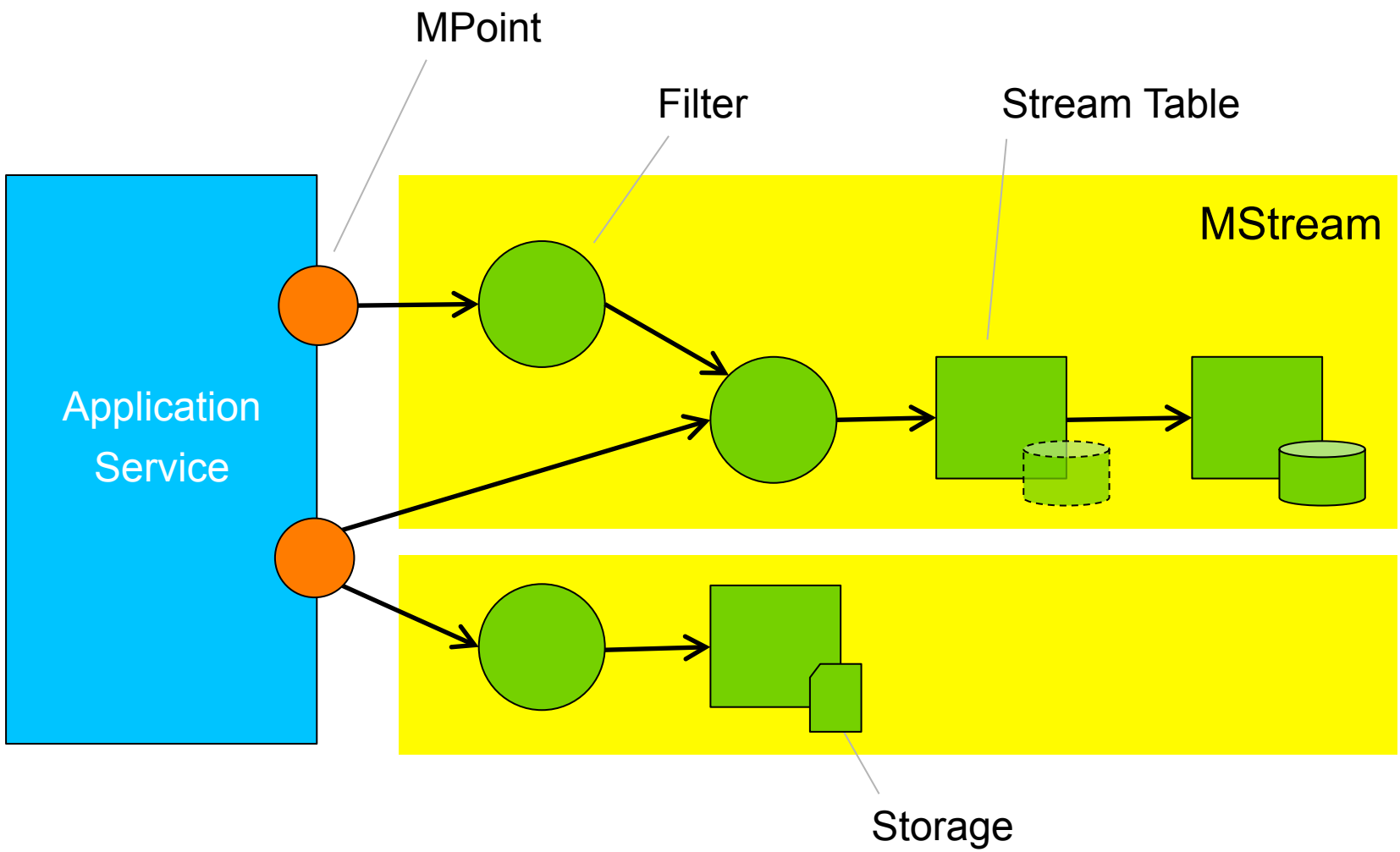
Goals of OML

- All experiment output in one place
- Capturing everything – most importantly meta data
- Separation of concerns
  - Instrumenting
  - Collecting
- Minimizing measurement collection overhead
  - Time
  - Traffic interference
- Support for steerable experiments
  - Access to data in different places

Concepts



Concepts



Defining MPoints

```
defApplication('system:app:otg') do |a|
    ...
    a.defMeasurement('channel') do |m|
      m.defMetric('size', :int)
      m.defMetric('speed', :float)
      ...
    end

  end
```

Defining MStreams

```
defGroup('g2') do |g|
    g.addApplication('system:app:otg') do |a|

      a.measure('channel', :samples => 10) do |m|
        m.metric 'size', :filter => 'avg'
      end

    end
  end
```

OML'ified Application

- Traffic Generation/Measurements
  - OTG … Traffic Generator
  - Iperf
- Monitoring
  - Libtrace
  - Libsigar
  - Spectrum Analyzer
  - GPS
  - (Weather)
- Components
  - TinyOS/Motes
  - (GnuRadio)

Filters

- Plug-in Architecture
- User extensibility
- Current List
  - Stddev
  - Average
  - First
  - Histogram

Building an OML app in 5 minutes

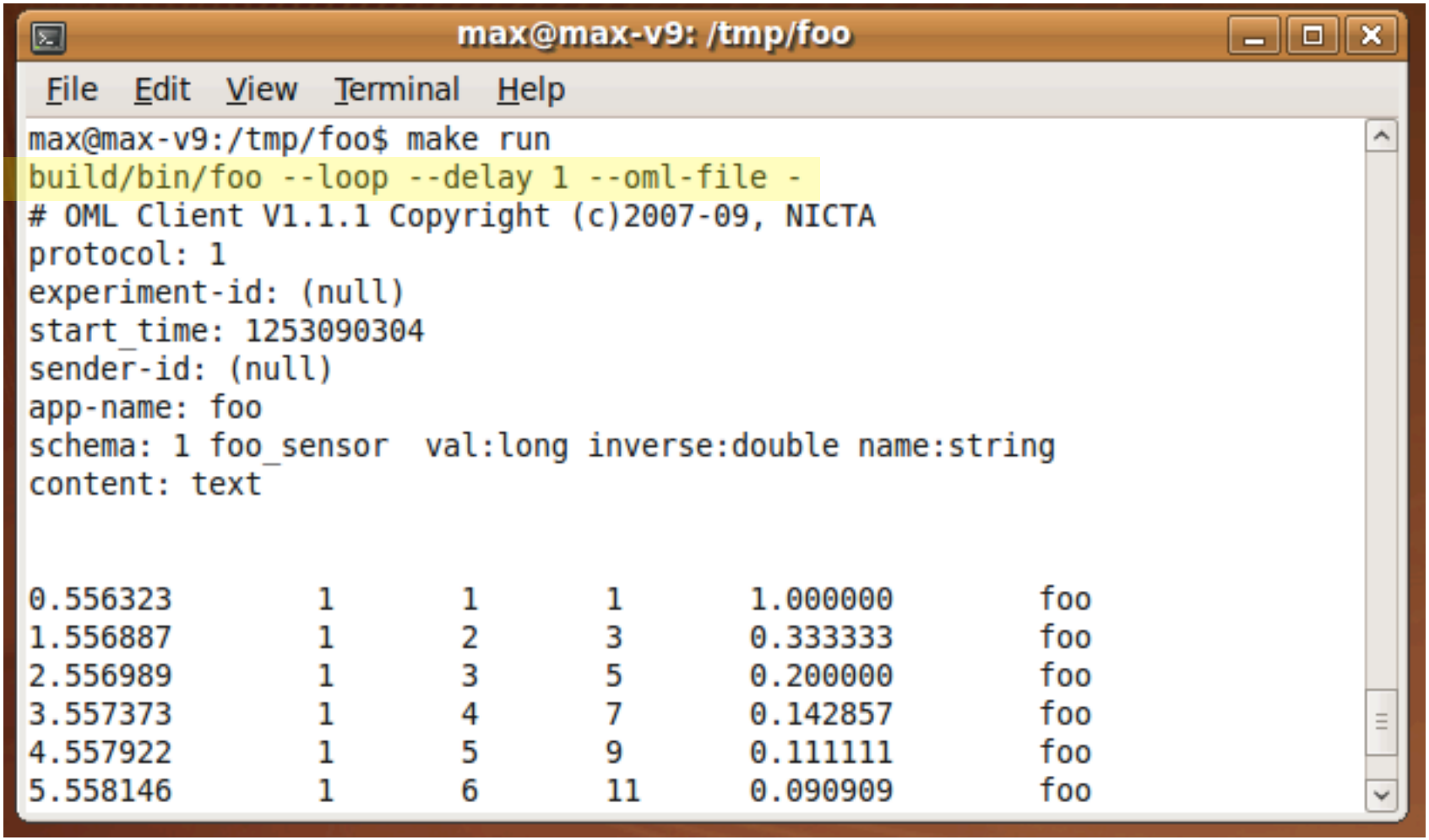
Building an OML app in 5 minutes

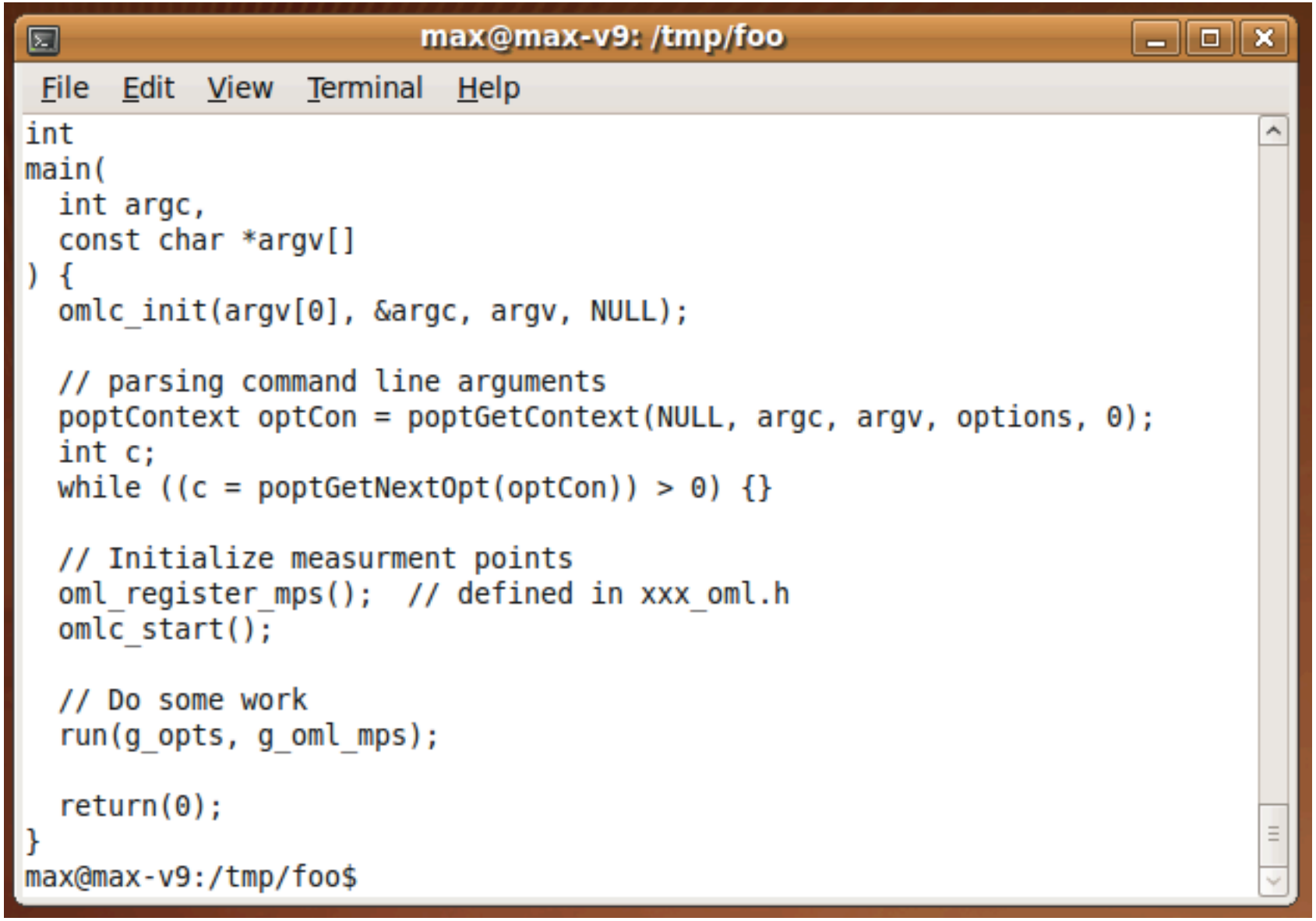
Building an OML app in 5 minutes

Building an OML app in 5 minutes

```
$ cat foo.app

defApplication('max:app:foo', 'foo') do |a|
  ...
  a.defProperty('loop', 'Create periodic result', ?l, ...
  a.defMeasurement("sensor") do |m|
    m.defMetric('val', 'long')
    m.defMetric('inverse', 'double')
    m.defMetric('name', 'string')
  end
  ...
end
```


max@max-v9: /tmp/foo
File Edit View Terminal Help
max@max-v9:/tmp/foo$ make run
build/bin/foo --loop --delay 1 --oml-file -
OML Client V1.1.1 Copyright (c)2007-09, NICTA
protocol: 1
experiment-id: (null)
start_time: 1253090304
sender-id: (null)
app-name: foo
schema: 1 foo_sensor val:long inverse:double name:string
content: text
0.556323 1 1 1 1.000000 foo
1.556887 1 2 3 0.333333 foo
2.556989 1 3 5 0.200000 foo
3.557373 1 4 7 0.142857 foo
4.557922 1 5 9 0.111111 foo
5.558146 1 6 11 0.090909 foo

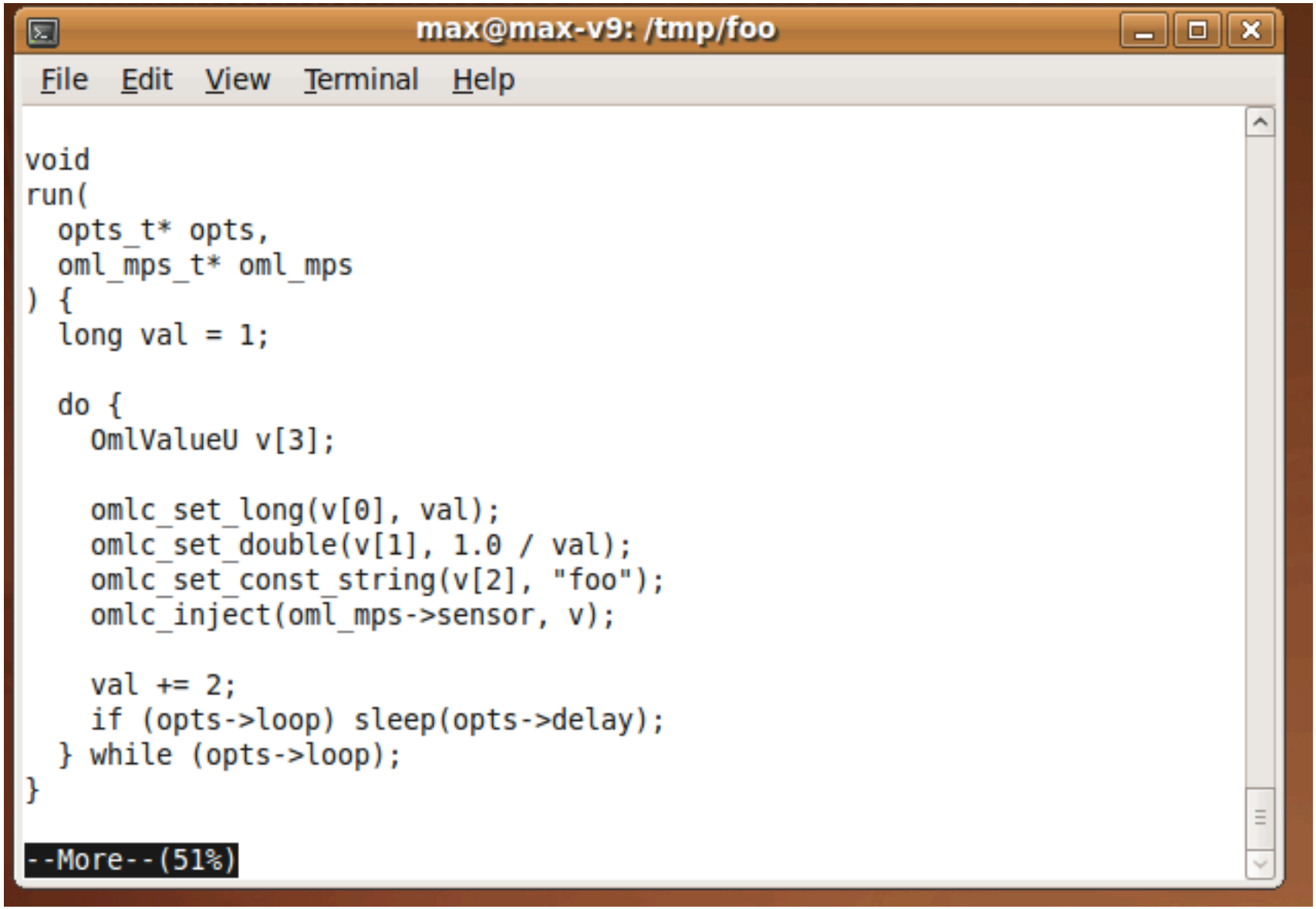

max@max-v9: /tmp/foo
File Edit View Terminal Help
int
main(
  int argc,
  const char *argv[]
) {
  omlc_init(argv[0], &argc, argv, NULL);

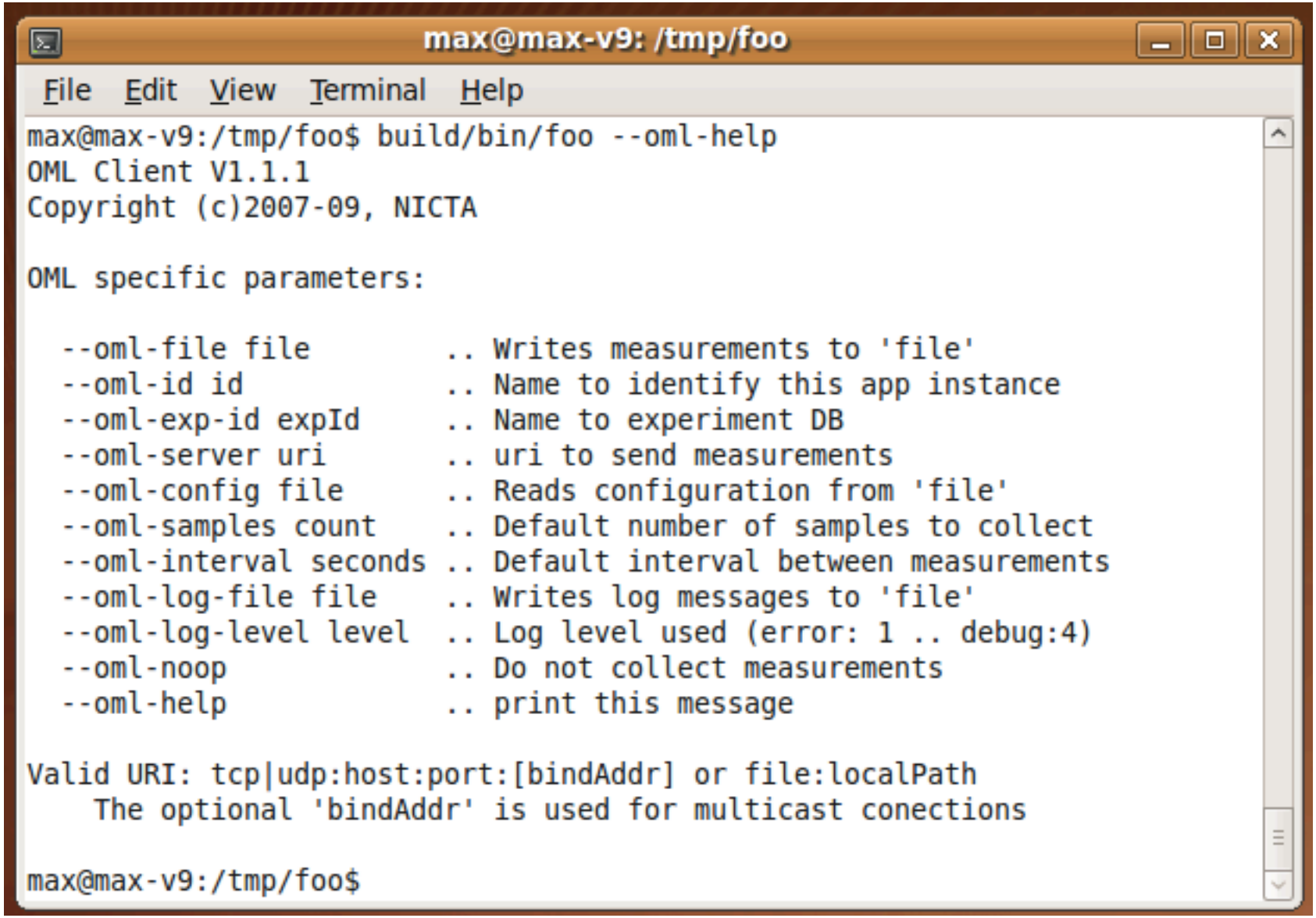
  // parsing command line arguments
  poptContext optCon = poptGetContext(NULL, argc, argv, options, 0);
  int c;
  while ((c = poptGetNextOpt(optCon)) > 0) {}

  // Initialize measurment points
  oml_register_mps(); // defined in xxx_oml.h
  omlc_start();

  // Do some work
  run(g_opts, g_oml_mps);

  return(0);
}
max@max-v9:/tmp/foo$


max@max-v9: /tmp/foo
File Edit View Terminal Help
void
run(
opts_t* opts,
oml_mps_t* oml_mps
) {
long val = 1;
do {
OmlValueU v[3];
omlc_set_long(v[0], val);
omlc_set_double(v[1], 1.0 / val);
omlc_set_const_string(v[2], "foo");
omlc_inject(oml_mps->sensor, v);
val += 2;
if (opts->loop) sleep(opts->delay);
} while (opts->loop);
}
--More--(51%)


max@max-v9: /tmp/foo
File Edit View Terminal Help
max@max-v9:/tmp/foo$ build/bin/foo --oml-help
OML Client V1.1.1
Copyright (c)2007-09, NICTA
OML specific parameters:
--oml-file file .. Writes measurements to 'file'
--oml-id id .. Name to identify this app instance
--oml-exp-id expId .. Name to experiment DB
--oml-server uri .. uri to send measurements
--oml-config file .. Reads configuration from 'file'
--oml-samples count .. Default number of samples to collect
--oml-interval seconds .. Default interval between measurements
--oml-log-file file .. Writes log messages to 'file'
--oml-log-level level .. Log level used (error: 1 .. debug:4)
--oml-noop .. Do not collect measurements
--oml-help .. print this message
Valid URI: tcp|udp:host:port:[bindAddr] or file:localPath
The optional 'bindAddr' is used for multicast conections
max@max-v9:/tmp/foo$

```
<omlc exp_id="exp99" id="node11">
  <sink url="file:-">
    <ms in=”sensor" interval="2" >
      <f on=”val"/> <!– default filter –>
      <f name="avg">
        <p name="key">3</p>
      </f>
    </ms>
  </sink>
</omlc>
```

Status

- http://omf.mytestbed.net/projects/show/oml
- MIT License
- 2009-09-11: Release of version 2.3
  - Support for re-starting existing experiments (long running)
  - Supports for text-based protocol for simple clients
  - Experimental API for implementing custom filters

Future

- Additional data types (IP, blob)
  - Potentially move to IPFIX
- Multi-dimensional data (spectrum, geographic – trip line)
- Triggers (Steerable)
- Resolve service integration vs. observation
- Streaming database
- (Distributed processing – map/reduce)
- ((Privacy))

OML Overview

Oct '09

Max Ott

NICTA

Australian Government

Department of Communications, Information Technology and the Arts

Australian Research Council

NICTA Members

NICTA Partners