JSAMP Java Toolkit

Mark Taylor (AstroGrid/Bristol)

Apps 1: SAMP
IVOA Interop, Baltimore

27 October 2008

`$Id: jsamp.tex,v 1.6 2008/10/27 19:48:31 mbt Exp $`

Outline

- Overview and Features
- Use of client library to SAMPify applications
- Use of tools for diagnostics etc
- Demo

JSAMP Overview

JSAMP is a SAMP toolkit written in Java

http://deployer.astrogrid.org/software/jsamp/index.html

It contains:

- Hub implementation
  - ▷ Includes client activity diagnostics (graphical and/or logging)
- Library classes for SAMPifying Java applications
  - ▷ Easy API access to hub functions
  - ▷ Hub connector automatically keeps track of hub presence and registered clients
  - ▷ Can take care of asynchronous send/receive
  - ▷ GUI utility classes to send messages, register, unregister, start hub . . .
- Tools for diagnostics, testing and debugging, useful for developers and users
  - ▷ `hubrunner`: Runs a hub, with optional graphical display and logging
  - ▷ `hubmonitor`: Displays registered clients, with metadata and subscriptions
  - ▷ `snooper`: Subscribes to some or all MTypes and logs message receipt
  - ▷ `messagesender`: Sends a message from the command line
  - ▷ `hubtester`: Tests JSAMP or third party hubs for compliance with the standard
  - ▷ `calcstorm`: Runs multiple clients simultaneously for benchmark purposes

JSAMP Features

- High quality implementation
  - Robust
  - Fully documented
  - Fully implements SAMP 1.0 WD
  - Easy to use low- or high-level SAMP facilities
- Pluggable architecture
  - For Standard Profile, choice of XML-RPC implementations available
    - ▷ Apache: based on Apache XML-RPC library v1.2-b1
    - ▷ Internal: no external libraries required
    - ▷ roll your own, based on your choice of third-party XML-RPC library
  - Ready for Profiles other than Standard Profile (XML-RPC) if required
- Easy deployment
  - Pure Java J2SE 1.4 or greater
  - Few dependencies (can run with no external libraries)
  - Open source
  - Unrestrictive license

Client Library

Convenience classes for `map`-like SAMP objects

- Includes `Metadata`, `Subscriptions`, `Message`, `Response`, `RegInfo`, . . .
- Correct usage enforced by library, avoids looking up in standard
  - ▷ `response.getErrInfo().getErrortxt()` replaces `(String) ((Map)response.get("samp.error")).get("samp.errortxt")`

Pluggable profile usage

- (Standard) Profile implementation kept separate from rest of package
- Can provide custom implementation of `ClientProfile` interface (factory for `HubConnection` objects)
- No non-*Standard* profiles defined yet, but good for testing anyway

Client Library

`HubConnection` interface

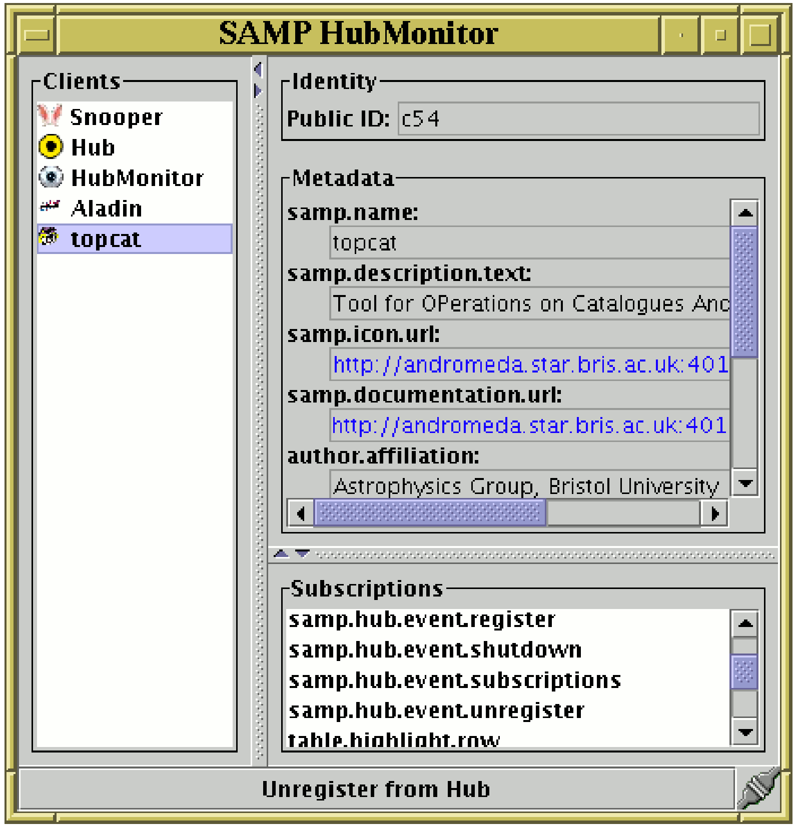
- Represents connection to a single running hub
- Interface almost identical to SAMP hub interface (SAMP doc Sec 3.11)
- Profile-dependent (XML-RPC, private ID) details hidden from calling code
- Adds `setCallable(CallableClient callable)` method

`HubConnector` class

- Higher level than `HubConnection`
- Keeps track of hubs starting/stopping
  - ▷ Watches out for new hubs starting up
  - ▷ Reacts when existing hub shuts down
  - ▷ Re-registers current metadata and subscriptions automatically on reconnection
- Maintains list of other registered clients and their metadata/subscriptions
- Provides synchronous façade for hub asynchronous call/response
- Facilitates provision of callback services (add/remove `MessageHandler`/`ResponseHandler`)
- Provides GUI facilities

Client Library GUI Components

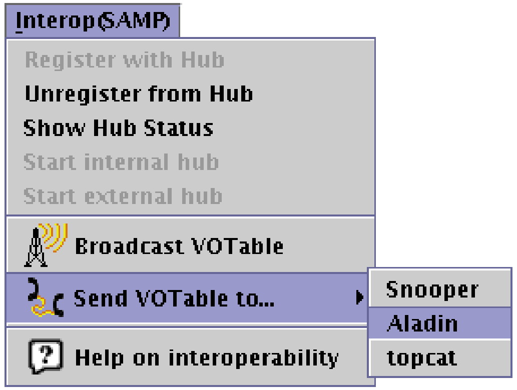
- ListModel of registered clients, updated automatically by hub events
- Automatic management of hub connection, keeping track of hub shutdown/startup
- Off-the-shelf components for viewing hub and client status



Public ID: c54

samp.name:
topcat
samp.description.text:
Tool for OPerations on Catalogues Anc
samp.icon.url:
http://andromeda.star.bris.ac.uk:401
samp.documentation.url:
http://andromeda.star.bris.ac.uk:401
author.affiliation:
Astrophysics Group, Bristol University

samp.hub.event.register
samp.hub.event.shutdown
samp.hub.event.subscriptions
samp.hub.event.unregister
table.highlight.row



- Actions to register and unregister with hub
- Actions to start internal/external hub
- Actions to broadcast given messages
- Submenus to send given messages to subscribed recipients

JSAMP Tools

JSAMP comes with several command-line tools

- Useful for diagnostics, testing, monitoring, . . .

```
% java -jar jsamp.jar

   This is JSAMP.

   JSAMP toolkit version:
      0.2-1
   SAMP standard version:
      SAMP WD 1.0 (2008-06-25)
   Author:
      Mark Taylor (m.b.taylor@bristol.ac.uk)
   WWW:
      http://deployer.astrogrid.org/software/jsamp/index.html

   Usage:
      org.astrogrid.samp.JSamp [-help] [-version] <command> [-help] <cmd-args>
      <command-class> [-help] <cmd-args>

   Commands (command-classes) are:
      hubmonitor      (org.astrogrid.samp.gui.HubMonitor)
      hubtester       (org.astrogrid.samp.test.HubTester)
      calcstorm       (org.astrogrid.samp.test.CalcStorm)
      messagesender   (org.astrogrid.samp.test.MessageSender)
      snooper         (org.astrogrid.samp.test.Snooper)
      hubrunner       (org.astrogrid.samp.xmlrpc.HubRunner)
```

HubRunner

hubrunner: Runs a hub

- Usage:

  ```
      org.astrogrid.samp.xmlrpc.HubRunner
         [-help] [-/+verbose] [-xmlrpc apache|internal] [-nogui]
  ```

- Optionally displays a window showing registered clients with metadata and subscriptions
- Optionally writes logging information to standard output for registrations, message etc
- Can choose XML-RPC implementation (apache, internal or custom)

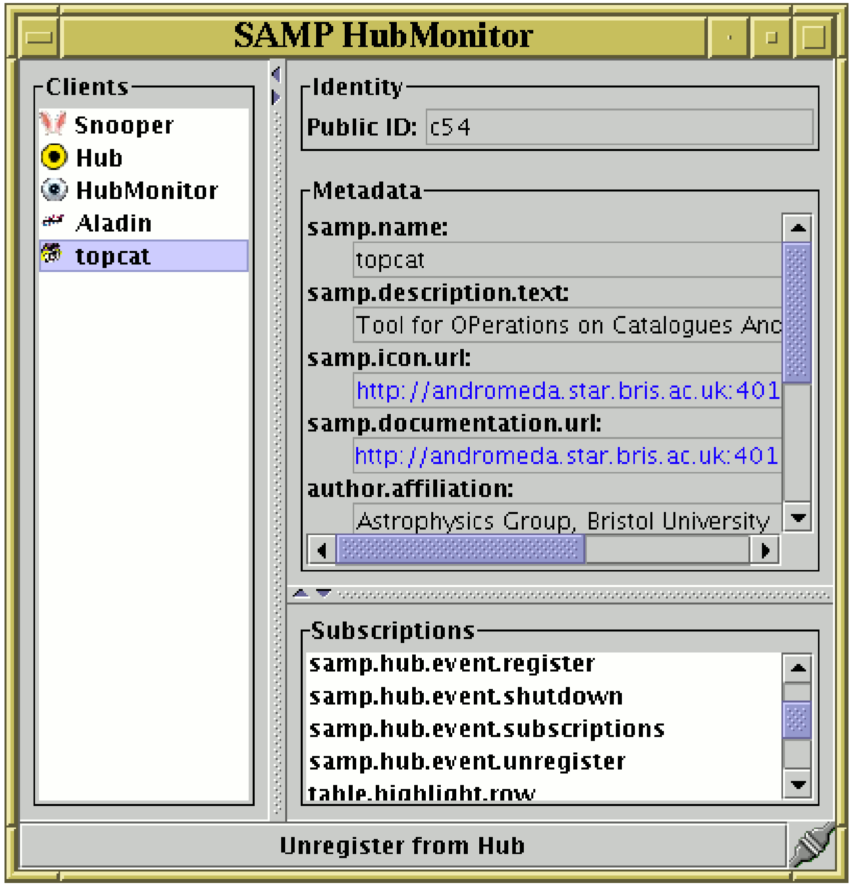
HubMonitor

hubmonitor: GUI display for client's-eye view of hub

- Usage:

```
org.astrogrid.samp.gui.HubMonitor
      [-help] [+/-verbose] [-xmlrpc apache|internal]
      [-auto <secs>] [-nogui]
```

- Shows all information which can be obtained from Hub status messages:
  - which clients are registered
  - declared metadata for each client
  - declared subscriptions for each client
- Automatically connects/disconnects as hubs start/stop
- Implementation is very thin wrapper around JSAMP client classes — serves as example client code



Snooper
Hub
HubMonitor
Aladin
topcat

Public ID: c54

samp.name:
topcat

samp.description.text:
Tool for OPerations on Catalogues Anc

samp.icon.url:
http://andromeda.star.bris.ac.uk:401

samp.documentation.url:
http://andromeda.star.bris.ac.uk:401

author.affiliation:
Astrophysics Group, Bristol University

samp.hub.event.register
samp.hub.event.shutdown
samp.hub.event.subscriptions
samp.hub.event.unregister
table.highlight.row

Snooper

snooper: Logs messages received from hub

- Usage:

```
    org.astrogrid.samp.test.Snooper
        [-help] [-/+verbose] [-xmlrpc apache|internal]
        [-mtypes <pattern>]
```

- Connects to hub, and writes details of all messages received to stdout
- By subscribing to MType pattern “*”, can report all broadcast messages
- . . . but no way for client to snoop on targetted (non-broadcast) messages
- Example:

```
% java -jar jsamp.jar snooper -mtypes '*'

hub (Hub) --- notify
   samp.mtype: samp.hub.event.register
   samp.params: {
      id: c141 }
hub (Hub) --- notify
   samp.mtype: samp.hub.event.metadata
   samp.params: {
      metadata: {
         samp.description.text: Rudimentary integer arithmetic application
         samp.name: Calculator
         samp.icon.url: http://www.star.bris.ac.uk/~mbt/plastic/images/tinycalc.gif }
      id: c141 }
```

MessageSender

`messagesender`: Sends a single message

- Usage:

```
org.astrogrid.samp.test.MessageSender
    [-help] [-/+verbose] [-xmlrpc apache|internal]
    -mtype <mtype> [-param <name> <value> ...]
    [-target <receiverId> ...] [-mode sync|async|notify]
    [-sendername <appname>] [-sendermeta <metaname> <metavalue>]
```

- Sends a single message from the command line to one or more clients
    - ▷ Client registers, sends message, waits for result(s), unregisters
- Useful for testing message receipt when no client is available which sends that MType
- Syntax can currently only cope with `string`-valued arguments (not `list`s or `map`s)
- Example:

```
% java -jar jsamp.jar messagesender \
            -mode async -mtype table.load.votable \
            -param url file://localhost/mbt/data/table/messier.xml

c5512 (topcat)
{samp.status=samp.ok, samp.result={}}
```

- Scripting languages may be more appropriate for this (e.g. SAMPy equivalent)

HubTester

`hubtester`: Tests a currently running hub

- Usage:

```
org.astrogrid.samp.test.HubTester
    [-help] [-/+verbose] [-xmlrpc apache|internal] [-gui]
```

- Attempts to test all aspects of hub behaviour as documented in SAMP standard
- Prints stacktrace on failure (mostly need to examine source code to understand error)
- Hubs which pass this should be in pretty good shape
  - ▷ Currently: JSAMP and SAMPy
- Useful for establishing hub implementation correctness, ensuring interoperability

CalcStorm

`calcstorm`: Runs multiple clients

- Usage:

```
    org.astrogrid.samp.test.CalcStorm
          [-help] [-/+verbose] [-xmlrpc apache|internal] [-gui]
          [-nclient <n>] [-nquery <n>] [-mode sync|async|notify|random]
```

- Registers *nclient* clients and gets them to pass *nquery* messages each to the others
- Messages are simple arithmetic queries from sender to recipient (custom MTypes)
- User can specify delivery pattern (Sync Call/Response, Async Call/Response, Notification) or mixture
- Checks that responses are as expected (none lost, return values correct)
- Useful stress test or benchmark for third party hub (and JSAMP client library)

Results (JSAMP hub, Standard Profile, Internal XML-RPC implementation):

- Works fine with 100 clients sending/receiving messages simultaneously
- Time per message ∼10 ms
- Should be quite fast enough for GUI-event-generated messages
- Different scenarios requiring high throughput may need new *Profiles*

Demo