TANGO CONTROL SYSTEM MANAGEMENT TOOL

P.Verdier, J.L.Pons, F.Poncet, ESRF, Grenoble, France
N.Leclercq, Soleil, Gif-sur-Yvette, France

Abstract

TANGO [1] is an object oriented control system tool kit based on CORBA initially developed at ESRF[2]. It is now also developed and used by Soleil, Elettra, Alba, Desy, MAX-Lab, FRM II and some other labs. The TANGO concept is a fully distributed object oriented control system.

That means that several processes (called servers) are running on many different hosts. Each server manages one or several TANGO classes. Each class could have one or several instances (call devices).

This poster will show existing tools to configure, survey and manage a large number of TANGO components.

GOAL

The first goal is to know at a quick glance, if everything is OK in a control system, and otherwise to be able to diagnose a problem and solve it.

The second goal is to configure the control system and its components.

The third goal is to have long term analysis on components (logs, statistics, usage,....)

PRINCIPLE

On each host to be controlled, a device server (called Starter[3]) takes care of all device servers running (or supposed to be running) on this machine. The controlled server list is read from the TANGO database. A graphical client (called Astor[4]) is connected to all Starter servers and is able to:

- Display the control system status and component status using coloured icons,.
- Execute actions on components (start, stop, test, configure, display information, …)
- Execute diagnostics on components.
- Execute global analysis on large number of crates or database.

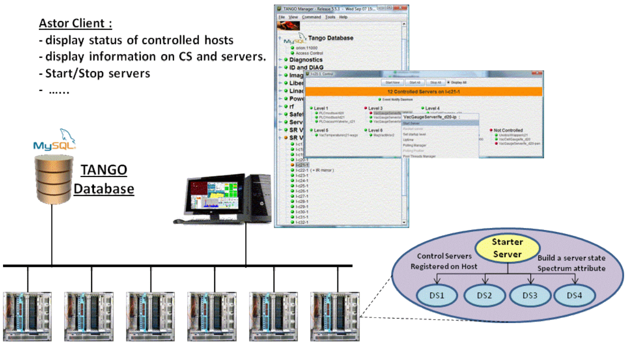

Astor Client :
- display status of controlled hosts
- display information on CS and servers.
- Start/Stop servers
- ……
MySQL
TANGO Database
TANGO Manager - Release 5.5.3 - Wed Sep 07 15...
File View Command Tools Help
Tango Database
orion:11000
Access Control
Diagnostics
ID and DIAG
12 Controlled Servers on l-c21-1
Level 1
Level 3
Level 4
Level 5
Level 6
Not Controlled
VacGaugeServerfe_d20-ip
Starter Server
Control Servers Registered on Host
Build a server state Spectrum attribute
DS1
DS2
DS3
DS4

Figure 1: Astor/Starter principle.

STARTER DEVICE SERVER

This Tango device server, written in C++, is dedicated to control system management. An instance of the Starter must be running on each host in the control system.

Controlled servers status

When a Tango device server starts, it registers itself automatically in the TANGO database as running on specified host. At start-up, the Starter takes this list of servers registered on the host .

Periodically, a system call gets the list of running processes. For each controlled server, the Starter checks if the server is the running processes list. If it is running a second check is done by a ping on the administrative device to know if the server is alive. And finally a list containing the server names and their status is built.

Starting Servers

For each server, a control level can be defined. Level 0 means not controlled. At start-up, the Starter takes a list of servers registered on the host and its control level. Starting at level 1 to level n, if the specified server is not already running, the Starter tries to start it.

In order to not overload the database, the servers are started sequentially. To start the next server, one of the following conditions must be met:

- Server is alive.
- Server has failed.
- Time out.

A server can be started at any time on a command from a client. Using the same manner, a client command is able to start all servers having the same control level.

It is possible to configure the starter, using a property, to re-start a server automatically if it fails. This property defines a minimum time that has be running, to ensure that this server is able to run a bit.

Implementation

The Starter device server is supported for 3 operating systems:

- Windows.
- Linux
- Solaris

One of the main problems has been the different API's for system calls. Mainly to get running processes.

ASTOR – CLIENT TANGO MANAGER

A graphical client, using the java swing frame containing a tree to display the control system status.

On top of the tree we can find:

- The database status
- The access control status (if activated).

FEATURE OVERVIEW

Control System Status in a Quick Glance

The host status read from the Starter device server is displayed on a tree using coloured icons.

- Green if all controlled servers are running
- Blue if at least one server does not respond (start-up phase)
- Orange if at leas one controlled server is stopped
- Red if the starter itself does not respond.

The hosts can be grouped under branches by usage, operating system or whatever.... The status of the hosts under a branch is reported to the branch node.

Executing actions on Tango device servers

A panel can be opened on a crate for more details. The status of each server is also displayed by a coloured icon.

- Green if the server is running
- Blue if the server is running but not responding.
- Red if the server is not running.

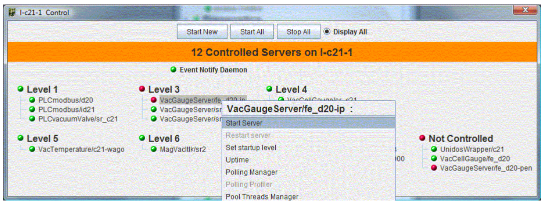


Figure 2: Host panel for server management.

By a right click on a server, a pop up menu is displayed to have an action or information on this server.

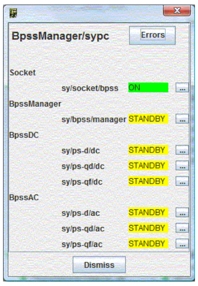
- Start/Stop servers.
- Test devices (single command or attribute).
- Check device states.

With one of the top buttons, you can start or stop all controlled servers registered on the specified host.

Using the same approach you can start a new device server. It will become a controlled server after you have chosen the control level.

On the host panel (figure 2), the displayed status, is the server status, and not the device status, as most of the people belong. The specified server is able to serve many devices, and a simple icon cannot represents a combination of several status.

To know the status of each device select the item “Check States” and a panel showing devices status, sorted by classes.

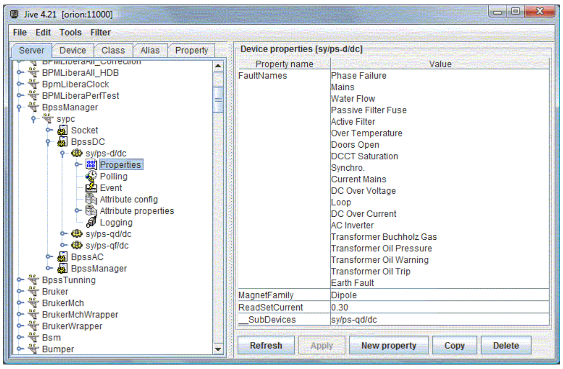


Socket		
	sy/socket/bpss	ON
BpssManager		
	sy/bpss/manager	STANDBY
BpssDC		
	sy/ps-d/dc	STANDBY
	sy/ps-qd/dc	STANDBY
	sy/ps-qf/dc	STANDBY
BpssAC		
	sy/ps-d/ac	STANDBY
	sy/ps-qd/ac	STANDBY
	sy/ps-qf/ac	STANDBY

Figure 3: Device status panel

Configure and Display Information

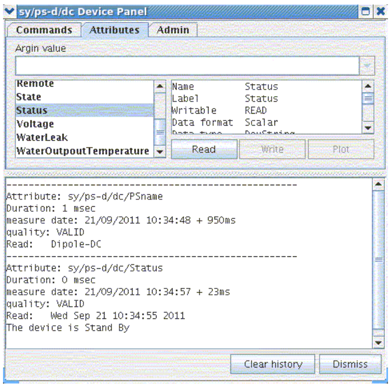
A database browser (called Jive) can be launched to edit device or class properties (see figure 4).



Property name	Value
FaultNames	Phase Failure Mains Water Flow Passive Filter Fuse Active Filter Over Temperature Doors Open DCCT Saturation Synchro. Current Mains DC Over Voltage Loop DC Over Current AC Inverter Transformer Buchholz Gas Transformer Oil Pressure Transformer Oil Warning Transformer Oil Trip Earth Fault
MagnetFamily	Dipole
ReadSetCurrent	0.30
__SubDevices	sy/ps-qd/dc

Figure 4: Jive (database browser).

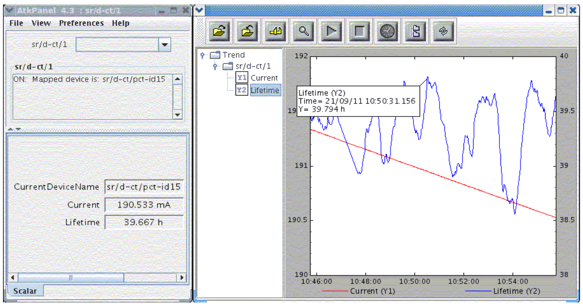
It is also used to have information as to where a server is running, on which port, uptime,... Or to add new components, test devices(see figure 5) and start a generic panel (called ATKpanel) to monitor devices (see figure 6).



```
Name         Status
Label        Status
Writable     READ
Data format  Scalar
```

```
-----------------------------------------------------
Attribute: sy/ps-d/dc/PSname
Duration: 1 msec
measure date: 21/09/2011 10:34:48 + 950ms
quality: VALID
Read:   Dipole-DC
-----------------------------------------------------
Attribute: sy/ps-d/dc/Status
Duration: 0 msec
measure date: 21/09/2011 10:34:57 + 23ms
quality: VALID
Read:   Wed Sep 21 10:34:55 2011
The device is Stand By
```

Figure 5: Device test panel



sr/d-ct/1

ON: Mapped device is: sr/d-ct/pct-id15

CurrentDeviceName	sr/d-ct/pct-id15
Current	190.533 mA
Lifetime	39.667 h

```
Lifetime (Y2)
Time= 21/09/11 10:50:31.156
Y= 39.794 h
```

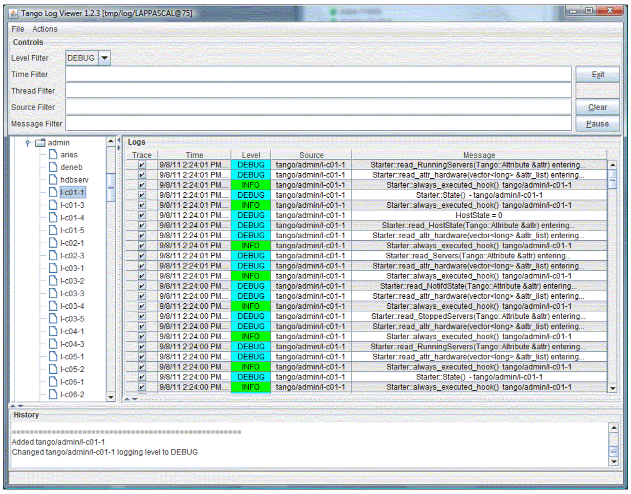
Figure 6: Device monitoring panel

A wizard is available to guide the user to create a new server and to set class and device properties

Jive is able to set the polling period for device attributes and to set event parameters.

Help to debug

To help debug a device, Jive and Astor propose a viewer (called LogViewer) to display on a specified device several outputs levels (see figure 7). These outputs are time stamped and can be filtered to improve analysis.



Trace	Time	Level	Source	Message
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_RunningServers(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:01 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::State() - tango/admin/l-c01-1
✓	9/8/11 2:24:01 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	HostState = 0
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_HostState(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:01 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_Servers(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:01 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:01 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_NotifdState(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:00 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_StoppedServers(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:00 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_RunningServers(Tango::Attribute &attr) entering...
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::read_attr_hardware(vector<long> &attr_list) entering...
✓	9/8/11 2:24:00 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1
✓	9/8/11 2:24:00 PM....	DEBUG	tango/admin/l-c01-1	Starter::State() - tango/admin/l-c01-1
✓	9/8/11 2:24:00 PM....	INFO	tango/admin/l-c01-1	Starter::always_executed_hook() tango/admin/l-c01-1

```
=====================================================
Added tango/admin/l-c01-1
Changed tango/admin/l-c01-1 logging level to DEBUG
```

Figure 7: LogViewer (device outputs)

Attribute Polling Configuration and Tuning

A Tango device attribute can be polled periodically. To configure this polling, a diagnostic tool is available. It displays when each attribute has been polled the last four times (see figure 8) and how much time was required to read and set it. It is very useful to tune the device attribute polling when a server has several devices with many attributes each. Or in case of a hardware problem with time out in reading.

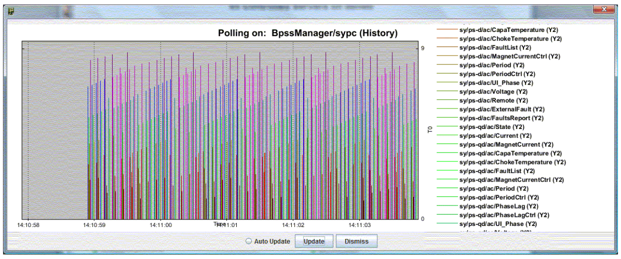


Figure 8: Polling profiler

By default, a single thread is started to poll attributes. In case of several devices, a pool of threads is available. Astor proposes a graphic tool to distribute device(s) by thread as shown in figure 9.

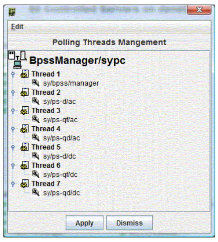
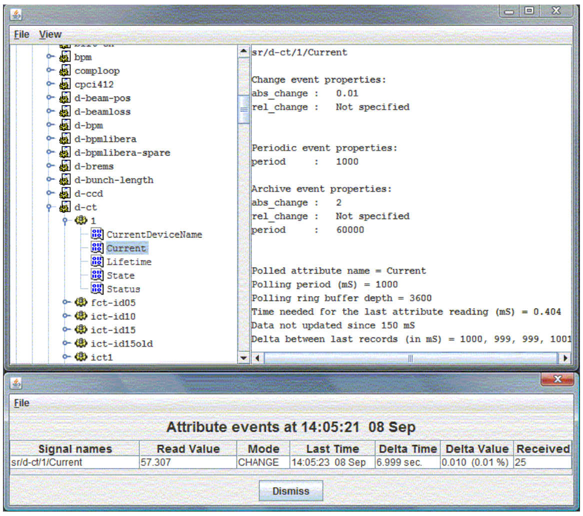


Figure 9: Pool of threads management

Events Configuration and Test

Tango communication between client and server, could be done on events. A client subscribes to on event (e.g. an attribute change value) and will be waken up when this event will be pushen by the device. To configure and test events , Astor proposes a tool able to browse database to select device attribute, configure event parameters (period, change value, …) and start a test to check if events are received as expected (see figure 10).



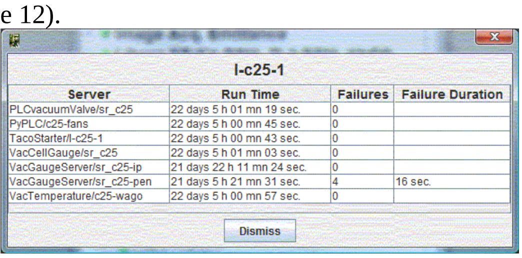
```
sr/d-ct/1/Current

Change event properties:
abs_change :   0.01
rel_change :   Not specified


Periodic event properties:
period     :   1000

Archive event properties:
abs_change :   2
rel_change :   Not specified
period     :   60000


Polled attribute name = Current
Polling period (mS) = 1000
Polling ring buffer depth = 3600
Time needed for the last attribute reading (mS) = 0.404
Data not updated since 150 mS
Delta between last records (in mS) = 1000, 999, 999, 1001
```

Attribute events at 14:05:21 08 Sep

Signal names	Read Value	Mode	Last Time	Delta Time	Delta Value	Received
sr/d-ct/1/Current	57.307	CHANGE	14:05:23 08 Sep	6.999 sec.	0.010 (0.01 %)	25

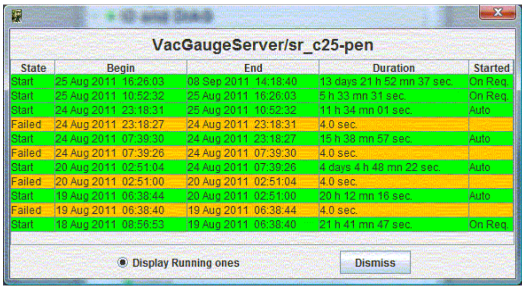
Servers Statistics

When a server, started by the Starter, is stopped by another way than a command on the Starter (kill signal, core dump, ...), this event is logged in a file. When the server is re started by the Starter, this information is logged too, Astor proposes to get information from this file for one host (see figure 11) or for all controlled hosts, and compute statistics an availability for servers (see figure 12).


l-c25-1

Server	Run Time	Failures	Failure Duration
PLCvacuumValve/sr_c25	22 days 5 h 01 mn 19 sec.	0	
PyPLC/c25-fans	22 days 5 h 00 mn 45 sec.	0	
TacoStarter/l-c25-1	22 days 5 h 00 mn 43 sec.	0	
VacCellGauge/sr_c25	22 days 5 h 01 mn 03 sec.	0	
VacGaugeServer/sr_c25-ip	21 days 22 h 11 mn 24 sec.	0	
VacGaugeServer/sr_c25-pen	21 days 5 h 21 mn 31 sec.	4	16 sec.
VacTemperature/c25-wago	22 days 5 h 00 mn 57 sec.	0	

Figure 11: Statistics on a host


VacGaugeServer/sr_c25-pen

State	Begin	End	Duration	Started
Start	25 Aug 2011 16:26:03	08 Sep 2011 14:18:40	13 days 21 h 52 mn 37 sec.	On Req.
Start	25 Aug 2011 10:52:32	25 Aug 2011 16:26:03	5 h 33 mn 31 sec.	On Req.
Start	24 Aug 2011 23:18:31	25 Aug 2011 10:52:32	11 h 34 mn 01 sec.	Auto
Failed	24 Aug 2011 23:18:27	24 Aug 2011 23:18:31	4.0 sec.	
Start	24 Aug 2011 07:39:30	24 Aug 2011 23:18:27	15 h 38 mn 57 sec.	Auto
Failed	24 Aug 2011 07:39:26	24 Aug 2011 07:39:30	4.0 sec.	
Start	20 Aug 2011 02:51:04	24 Aug 2011 07:39:26	4 days 4 h 48 mn 22 sec.	Auto
Failed	20 Aug 2011 02:51:00	20 Aug 2011 02:51:04	4.0 sec.	
Start	19 Aug 2011 06:38:44	20 Aug 2011 02:51:00	20 h 12 mn 16 sec.	Auto
Failed	19 Aug 2011 06:38:40	19 Aug 2011 06:38:44	4.0 sec.	
Start	18 Aug 2011 08:56:53	19 Aug 2011 06:38:40	21 h 41 mn 47 sec.	On Req.

Figure 12: Statistics on a server

Miscellaneous

Astor propose also:

- Simple logging about start/stop servers
- A panel to send command on several servers.
- A panel to show server usage (useful for large number of instance).
- A panel to configure control access if it is activated.

CONCLUSIONS

The couple Starter/Astor and related tools, are very useful to manage a large number of servers running on several hosts distributed around an accelerator or a large experimental physic control system. They help to configure diagnose problems, and for doing analysis and statistics.

REFERENCES

[1] http://www.tango-controls.org/

[2] http://www.esrf.eu/

[3] http://www.esrf.eu/computing/cs/tango/tango_doc/tools_doc/astor_doc/index.html

[4] http://www.esrf.fr/computing/cs/tango/tango_doc/ds_doc/tango-cs/System/starter/