Overview of TANGO Control system

EUROPEAN SYNCHROTRON RADIATION FACILITY
A simple acquisition system

• For simple system
• Many tools exists
  – Labview
  – Scadas
  – Etc...
• Bigger systems need several computers
A distributed system

- Need inter-computer communication
- Heterogenic system (languages, machines...)
- Serialization/de-serialization
- Variable addressing, topology
Object oriented

• Analyze a system as a set of objects communicating together
• Each object is in charge of its own life
• An object can be built on other objects
A European Control System framework

«Remote control anything and everything»

- A mature Open Source product
- > 100 man years of development
- Used in 15+ big instruments
- Active community

- Object oriented, topology independent, highly scalable,
- Operating system independent
- Multi language
TANGO is HIGHLY scalable!

From small embedded platforms
• ... to big scientific instruments
• > 1,000,000 signals and actuators
A software bus

Generic Services

- Data Analysis
- Config
- Monitor
- Sequencing
- Archiving

Application Tool-Kit

User environment
Matlab, Labview, Igor, Python

Development tools

TANGO Software Bus distributed on a network

API Library

Interface Generator

Catalog of device servers

Hardware

June 2010

JM Chaize, ESRF/CERN control workshop
Topology independent

- An element is identified on the overall network
- The computer is just a container
- The system is seen globally
- A database is used to localize the objects
The fundamental brick of TANGO is the **Device**

The device is a remote object

Standard interface for hardware equipment or service

Device class

Interface

HW

(motor...)

TANGO software bus
Some device(s)
A sophisticated device (RF cavity)
TANGO devices

Example: motor interface:

**MOTOR:**

- **Commands:** On(), Off(), …
- **Attributes:** Speed, Position
- **State:** On, Off, Alarm, Fault

**Hardware control code**

- Interface
  - Automatic code generator
  - To be written by the programmer
TANGO devices

- 1 Device can also interface complex systems
  - Hierarchical structure

![Diagram of TANGO devices with a hierarchical structure showing a Macro device: e.g. Accelerator and sub devices: e.g. powersupplies, ADC, modbus... connected to the TANGO Software Bus]

Client

Macro device: e.g. Accelerator

sub devices: e.g. powersupplies

sub devices: e.g. ADC, modbus...
Managing complexity simply

- Hierarchical structures ideal for managing complex systems

Diagram:
- GUI
- High Level SSA
- Sequencer
- Python Sequence
- Low Level RF
- 280V PS
- Wave Guide Switches
- 4x
- SSA
- Driver
- Dummy Load
- ADC Measurements
- Interlocks fast
- Interlocks slow
TANGO as a bridge

Object oriented layer **TANGO Software Bus** above EPICS...

Build a TANGO object from a set of channels
Commands & Attributes

- On the network a Tango device mainly has
  - **Command(s):** Used to implement “action” on a device (switching ON a power supply)
  - **Attribute(s):** Used for physical values (a motor position, a temperature, a spectrum, an matrix)
- Clients ask Tango devices to execute a command or read/write one of its attributes
- A Tango device also has a **state** and a **status** which are available using command(s) or as attribute(s)
Commands

- A command may have one input and one output argument.
- A limited set of argument data types are supported
  - Boolean, short, long, long64, float, double, string, unsigned short, unsigned long, unsigned long64, array of these, 2 exotic types and State data type
Attributes

- Self describing data via a configuration
- Thirteen data types supported:
  - Boolean, unsigned char, short, unsigned short, long, long64, unsigned long, unsigned long64, float, double, string, state and DevEncoded data type
- Three accessibility types
  - Read, write, read-write
- Three data formats
  - Scalar (one value), spectrum (an array of one dimension), image (an array of 2 dimensions)
Attributes

- When you read an attribute you receive:
  - The attribute data (luckily...)
  - An attribute quality factor
    - ATTR_VALID, ATTR_INVALID, ATTR_CHANGING, ATTR_ALARM, ATTR_WARNING
  - The date when the attribute was acquired by the server (number of seconds and usec since EPOCH)
  - Its name
  - Its dimension, data type and data format

- When you write an attribute, you send
  - The attribute name
  - The new attribute data
TANGO devices

- 1 Device can also interface complex systems
  - Bridge to other protocols

Build TANGO device from a set of OPC TAGs

OPC

Interface

Device

Air Conditioning

TANGO Software Bus

Infrastructure
TANGO devices

• 1 Device can also interface complex systems
  – Bridge to other protocols

TANGO Software Bus

Interface

Device I/O

Modbus TCP/IP

Embedded system
TANGO devices

- 1 Device can also interface complex systems
  - Bridge to other protocols

Build a TANGO device from DataSocket urls

DataSocket
Embedded TANGO servers

- Next steps

TANGO client

TANGO Software Bus

TANGO server
Mapped into a FPGA
Embedded TANGO servers

• Next steps

TANGO client

TANGO Software Bus

Gumstix SBC
The Tango Device Server

A Tango device server is the process where the Tango class(es) are running.

```
Device sr/v-ip/1
Device sr/v-ip/2
Device id4/mot/1
Device id4/mot/1
Device id4/mot/3
```

“ps” command shows one device server
Synchronous
TANGO Communication

- Asynchronous
TANGO Communication

• Event Driven
Standardise interfaces of equipment of the same types

Abstract interface classes

- Abstract motor
  - Concrete Implem
- Abstract powersupply
  - Concrete Implem

Different types of motors
Different types of powersupplies
Much more than a software bus

- Code generator for C++, Java, python
- Configuration tool
- Administration tool
- Archiving service
- Access control service
- Logging service
- Scan service
- GUI Toolkit for Java, QT, Python
- Synopsis animation tool
- Alarm service
- Web interface
- Android support
- Tutorials

- Bindings
- Many utility classes
- Hardware access class catalog
- Mailing list support
How to try it?

• TANGO is available free of charge

• Source code distribution
  Downloadable on http://www.tango-controls.org/download

• Binary Packages
  – Available for Ubuntu + Debian Linux in the standard distributions
    `sudo apt-get install tango-common tango-db python-pytango`
  – Available for Windows on http://www.tango-controls.org/download

• The Tango Box
  – An Ubuntu virtual machine with Tango installed and configured for easy testing
  – Runs on VMware and Virtualbox
Use cases

- TANGO was born in particle accelerator world, however ...
- TANGO is a generic solution for any collection of objects
- Used also in other scientific domains like
  - Wind tunnels e.g. ONERA
  - Neutron source experiments (FRM2, Julich in Germany)
- Large laser installation
  - LMJ, PETAL, APOLLON
  - Free Electron Laser (Fermi)
- Small instruments
  - Thomx (art and medicine)
- Small installations
  - Embedded systems
  - Industrial supervision
TANGO - Generating economical activity

Instrumentation hardware market

- Tango compatible hardware
  - Supplier provides and guaranty The TANGO interface to their product

- Advantage for the supplier:
  - Sales argument for addressing TANGO community
  - Provide remote control of device
  - Profit for free from TANGO framework

- Advantage for the community/client/user
  - Easier integration
  - Better matching of the hardware features
Recent projects started with TANGO
Jdraw: Generic Synoptic animation
Jive

Database browser and Test Device Launcher
POGO Device Server Code Generator

```cpp
void Hazemeyer::write_Current(Tango::Atttribute &attr)
{
    DEBUG_STREAM << "Hazemeyer::write_Current(Tango::Atttribute &attr) entering."
}

void Hazemeyer::off()
{
    DEBUG_STREAM << "Hazemeyer::off(): entering...!" << endl;
    // Add your own code to control device here
}

void Hazemeyer::on()
{
    DEBUG_STREAM << "Hazemeyer::on(): entering...!" << endl;
    // Add your own code to control device here
}
```

Device Attributes Description

Hazemeyer Class

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Data Type</th>
<th>RW Type</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>DEF.DOUBLE</td>
<td>READ_WRITE</td>
<td>No</td>
</tr>
<tr>
<td>Voltage</td>
<td>DEF.DOUBLE</td>
<td>READ</td>
<td>No</td>
</tr>
<tr>
<td>Frequency</td>
<td>DEF.DOUBLE</td>
<td>READ</td>
<td>Yes</td>
</tr>
<tr>
<td>CurrentSetPoint</td>
<td>DEF.DOUBLE</td>
<td>READ</td>
<td>No</td>
</tr>
</tbody>
</table>
ATKpanel: a generic client
Astor/Starter
Tango Control System Manager

- Get host list
- Get status
- Get server list
Thank you for your attention

- WWW sites for TANGO
  - Common site [http://tango-controls.org](http://tango-controls.org)