TANGO COLLABORATION NEWS

J. Meyer (ESRF) on behalf of the Tango community
ALBA, DESY, ELETTRA, ESRF, SOLEIL

Abstract

During the last years, the Tango collaboration was and is still growing. More and more users are requesting new features and developing new tools for Tango. Decisions whether the requested features will be implemented and whether new tools will be part of the Tango distribution need to be made. The organizational aspects of the collaboration need to be clarified as well as the decision making process for new developments.

This paper will explain the collaboration, its organization and the decision making process as well as the latest facts and features around Tango.

Some ongoing developments are the new code generation tool to allow inheritance in the Tango class structure, the new event system for high bandwidth event distribution and the Tango packaging to allow installation with a few clicks.

WHAT IS TANGO?

Tango [1] is a control system tool kit developed by a community of institutes. It is object oriented with the notion of devices (objects) for each piece of hardware or software to be controlled. Tango classes are merged within operating system processes called Device Servers. Three types of communication between clients and servers are supported (synchronous, asynchronous and event driven).

But Tango is not only the software bus which handles the communication between device servers and clients. The Tango tool chain offers software from the hardware interface to the graphical user interface for several programming languages.

Tango utilities are available, with the basic installation, for code generation, device configuration and testing and for administration and survey of a whole Tango control system.

An archiving and a configuration snapshot system usable with Oracle or MySQL are also available.

Table 1 : Available Tango Modules

Module	Description
Core Libraries	Client/Server communication libraries for C++, Python and Java
Device Classes	About 300 hardware interface classes are available to download [1]
GUI Frameworks	Available for C++ and Python using QT, for Java using Swing and a web interface written in PHP
Client Bindings	LabView, Matlab and IgorPro
Tools	Pogo – Code generator for device classes in C++, Python and Java Jive – Configuration and testing tool Astor – Administration and survey of the Control system
Archiving	Archiving and snapshot system with GUIs and web interface. Usable with Oracle and MySQL
Alarm System	Event driven alarm service
Sardana	Framework for experiment control : Interface standardization, configuration, sequencing, command line interface

COLLABORATION HISTORY

Tango development started in 1999 at the ESRF. SOLEIL joined as the first partner in 2002, ELETTRA and ALBA joined in 2004 and the DESY (beamline controls) in 2008.

For every new member a new memorandum of understanding was signed by all collaboration partners.

We meet twice a year to discuss all ongoing projects. In case of lack of consensus, we tried to find a solution, all collaboration partners could agree upon.

A coordinator was named in each institute for all organisational, but also technical requests concerning Tango.

A mailing list is available for all questions and propositions to the whole Tango community.

A GROWING COMMUNITY

Since last year we have two new institutes requesting to join the collaboration: MAX-lab in Sweden, FRM-II in Germany. Tango is also used by other laboratories, for example LMJ (beam diagnostics) in France. Industrial companies are evaluating Tango, due to outsourcing requests from new projects.

The number of software development projects around Tango is increasing. To package the system and to keep the source repositories clean, we have to decide which projects will be part of the Tango distribution and which ones will be add-ons.

With the growing community, the increasing number of users and the foreseeable number of new developments around Tango, we have to find a new organisational form, to be sure, to take decisions on development priorities and strategies within a reasonable delay.

THE NEW ORGANIZATION

Taking into account the increasing number of users, we will reduce the frequency of Tango meetings to reduce organizational effort and cost. Specialised meetings on particular development projects are encouraged.

To allow a fast decision making process we are changing the collaboration management structure. The new structure has three levels:

1. The executive committee:

The executive committee takes the strategic decisions about developments in the Tango collaboration. There is one member from each institute who has signed the memorandum of understanding. This representative should have enough power to decide on allocating resources to develop software for Tango.

The committee will meet, at least, just after the Tango collaboration meetings.

2. The collaboration coordinator:

The collaboration coordinator is the central point of the organizational structure and liaises between the project leads and the executive committee. His responsibility is:

- To organize and coordinate the executive committee meetings, to produce a report of the committee meeting and to give feedback to the Tango community.
- To maintain a global project plan, in collaboration with the project leads, including requirements, schedule and resource requirements.

3. The project leaders

Besides the Tango core libraries, several packages are considered to be part of the Tango controls system. A list of these packages is maintained up-to-date by the collaboration coordinator and any change to this list is decided by the executive committee. Each package which is part of the Tango core has to have a project leader.

For the Tango community, the project leader is the contact person for all questions and remarks concerning that particular project. He is in charge of following the project schedule and ensuring the requirements are satisfied. In case of problems impacting on other Tango project(s), the project leader refers questions to the collaboration coordinator and eventually to the executive committee.

We distinguish two different collaboration membership types:

- **Committer**: must contribute resources to the collaboration. He is responsible for one or more Tango core packages.
- **Contributor**: can propose code modifications to the committers for the Tango core packages and submits Tango device classes to the public device classes repository.

How to accept a new collaboration partner?

To be an official member of the Tango collaboration, a new institute needs to sign the memorandum of understanding. New members are to be accepted with a unanimous decision by the executive committee.

How to get an official Tango decision?

All requests for decision should be sent to the collaboration coordinator. They will be presented to the executive committee during the next committee meeting.

Decisions are made by voting. The vote of each executive committee member is weighted according to its status as contributor or committer (cf. above). Each committee member has at least a weight of one. An extra vote is acquired if the committee member represents an institute which is also a committer.

ON-GOING PROJECTS

The Packaging

To allow an easy way to install and run Tango we prepare binary packages on top of the source code distribution.

A binary package is available for Windows, since a long time, from the Tango web site [1]. Now a first version of binary packages is available for Debian and Ubuntu Linux users. From Launchpad the different packages can be installed as needed [2]. Investigations are ongoing how to support binary packages for other Linux distributions.

The Tango Box

The Tango box is a virtual Linux computer which runs in the VMware Player [3] virtualisation software. On this virtual machine runs a Tango system and most of the Tango tools are installed and ready to be used. It offers an overview of the Tango software on a running system without installing Tango on a local machine. The software on the Tango box is updated once a year.

GUI Developments

A lot of effort is spent to add more features and new functionality to the available graphical toolkits.

The Python toolkit Tau supports the whole spectrum of viewers now. With the C++ toolkit QTango, synoptic displays can be created from CAD drawings. On the Java side, an on-going development will open the toolkit for different data sources. This should allow the usage of widgets with non Tango data sources.

Pogo the Code Generator

All the Tango classes follow the same skeleton. Therefore, a code generator (Pogo) has been written to generate these skeletons. This tool was available at the very beginning of Tango. Pogo was implemented using hand written parsing techniques. The decision was taken to re-write the code generator.

The new release of Pogo is based on modern techniques using Xtext [4] to create a Tango DSL (Domain Specific Language). This DSL is then used to describe the new Tango class. Using Xpand [5] and a set of templates, the Tango class skeleton is generated. Xtext and Xpand are part of the Eclipse modeling project [6].

With this new way of generating code, it is now possible to implement inheritance of Tango device classes properly. Only the inheritance of abstract interface classes was allowed before.

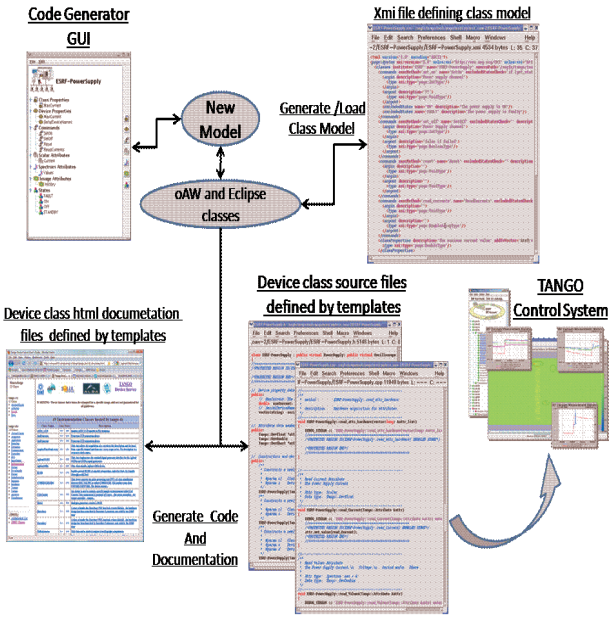

Figure 1 : Code generation with Pogo

The new code generator is available for generating C++ classes. The templates for Python and Java still need to be written.

THE NEAR FUTURE

A Faster Event System

The Tango event system is based on the CORBA [7] notification service, the implementation used is omniNotify [8]. Today's event rates are sufficient but cannot be improved due to the implementation of omniNotify (dead project). The detailed problems have been already described at ICALEPCS 2009 [9].

Performance measurements for event distribution have been carried out using the Data Distribution Service (DDS) [10] implementation OpenSplice [11] and the publisher/subscriber pattern of the ØMQ [12] Socket API.

The measured performance values are in received events per second between two machines (P4, 2.5GHz, Ubuntu 9.04 – Core 2 Duo, 2.6GHz, Ubuntu 9.04) on a 100 Mbit network.

Table 2 : Event System Performance Tests

Sub	1 int (32bits)			1024 int		
	Tango	**DDS**	**ØMQ**	**Tango**	**DDS**	**ØMQ**
1	770	12500	45000	650	1850	2400
5	400	7900	14000	200	1800	500
10	220	6500	7300	100	1700	230

DDS showed the best performance, for a growing number of subscribers, due to its multicasting protocol. But it has a set of drawbacks for programming and configuration. The ØMQ performance was measured only with unicast transmission because the multicasting showed reliability problems.

Table 3 : Event Systems Advantages and Drawbacks

	DDS	**ØMQ**
+	CORBA ORB/DDS cohabitation, Performance, QoS, Multicasting	No extra processes, Single cast performance, Can switch from uni- to multicast transmission
-	Three processes + shared memory per host, SIGKILL forbidden, No core dump, No dynamic data partitioning possible	Multicasting not yet 100% reliable, Young product, More integration code to write

The Tango philosophy is to keep it simple. ØMQ seems to be more adapted for us, even if the programming effort is higher and we have to collaborate with the implementers to make multicasting reliable. Due to the complexity of a multicasting set-up we would like to keep unicasting as the default transmission for the event system. But, multicasting should be available when needed.

Library for Image Acquisition (LIMA)

LIMA is a project for the unified control of two dimensional detectors. The aim is to clearly separate hardware specific code from the common software configuration and common features, like setting standard acquisition parameters (exposure time, external trigger, etc), file saving and image processing.

Requirements and specifications are actually collected from the interested institutes.

On top of the functionality of this library, a common Tango interface for 2D detectors should be defined.

REFERENCES

[1] http://www.tango-controls.org
[2] https://launchpad.net/~abogani/+archive/tango
[3] http://www.vmware.com/products/player
[4] http://www.eclipse.org/Xtext
[5] http://wiki.eclipse.org/Xpand
[6] http://www.eclipse.org/modeling
[7] http://www.corba.org
[8] http://omninotify.sourceforge.net
[9] E.Taurel, "Tango Kernel Status and Evolution", ICALEPCS'09, Kobe, Japan, 2009, THA001, p. 630 (2009); http://www.JACoW.org.
[10]http://www.omg.org/tlogy/documents/dds_spec_catalog.htm
[11] http://www.opensplice.org
[12] http://www.zeromq.org