TANGO collaboration and kernel status

- Brief introduction
- What's new since Icalepcs 2009
- New projects
- Re-thinking the Tango event system
What is Tango?

• An object oriented control system based on CORBA

• Each piece of hardware or software to be controlled (from the simplest to the most sophisticated) is a device

• A device is an instance of a Tango class which is hardware/software specific

• Device supports commands (actions) and attributes (data)
What is Tango?

- Tango class(es) are merged in operating system process called Device Server.

- Device configuration parameters and network address stored in a database.

- 3 types of communication:
  - Synchronous, Asynchronous and Event driven.

- 3 languages: C++, Python and Java.

- Collaboration between several institutes.
What's new since 2009?

- 3 kernel libraries releases
  - Tango 7.1.1 (11/2009)
    - Minor changes and bug fixes
  - Tango 7.2 (10/2010)
    - Thread safety on client part
    - Much faster algorithm used during device server process shutdown
    - Applications are able to subscribe to the same events several times
    - Minor changes and bug fixes
  - Tango 7.2.6 (03/2011)
    - Minor changes and bug fixes
What's new since 2009?

- Packaging
  - Linux binary distribution available
    - Based on Debian packaging system
    - 2 source packages
    - 19 binary packages (including documentation and debug packages)
    - Packages available from a Launchpad Tango-controls Personal Package Archive (PPA)
    - Starting with Ubuntu 11.10, packages available from Ubuntu Software Center
What's new since 2009?

- **Graphical User Interfaces**
  - Python GUI for Tango: Taurus
    - Based on PyQt 4
    - Talk WEAAUST 01 (Wednesday)
  - The C++ GUI (QTango) is now in its release 4 (also based on Qt 4)
    - Poster WEPKS 022 (Wednesday)
  - New widgets added to the ESRF Java GUI (ATK)
  - A newcomer: Comète
    - Java GUI supporting several data sources (not only Tango objects)
    - Poster / Mini oral WEMAU 012 (Wednesday)

- Code generator (Pogo) release 7 based on DSL using Xtext is now routinely used to generate C++ Tango class
What's new since 2009

• Collaboration Management
  • A new Memorandum Of Understanding (MoU)
  • 3 types of collaborators institutes
    • User (Not signing the MoU)
    • Contributors
    • Committers
  • Executive Committee (EC)
    • 1 member for each institute which has signed the MoU
    • Decision made by voting
      • Weight of 1 for each committee member plus 1 extra for each committers institute
  • Executive committee meeting organized at each Tango meeting
On-Going projects

• Java Tango classes and device server
  • Soleil has started an ambitious project of re-factoring and updating this part of the Tango kernel

• Software quality
  • Continuous Integration with Jenkins
    • 20 libraries flavor, 10 device server, 5 test suite
  • Improve the test suite
    • CxxTest selected as testing framework to unify the different test suite we have today
    • Increase test coverage to 75 %
The today's event system

• Based on the CORBA Notification service
  • omniNotify implementation

• Advantages
  • Simple on the event publisher side (no care about number of subscribers)

• Drawbacks
  • Require one extra process per host
  • Unicast network transfer
  • Use of CORBA Any objects
  • In some cases, large memory consumption
  • omniNotify is a dead project!
The new event system

• Based on ØMQ
  • http://www.zeromq.org

• What is ØMQ?
  • ØMQ looks like an embeddable networking library but acts like a concurrency framework. It gives you sockets that carry whole messages across various transports like in-process, inter-process, TCP and multicast. You can connect sockets N-to-N with patterns like fanout, pub-sub, task distribution and request-reply. Its asynchronous I/O model gives you scalable multicore applications, built as asynchronous message-processing task.

• Runs on most operating systems
• LGPL
The new event system

- 0MQ does not provide data encoding / decoding

- For synchronous communications, Tango uses CORBA Common Data Representation (CDR)

- ORB's compiler generates methods to encode / decode data to / from CDR

- Most of the event data are encoded using CORBA CDR and transported using 0MQ.
The new event system

- Transferred data between event publisher and subscriber:
  - String describing the event:
    - Fully qualified Tango attribute name plus the event type
  - A single byte encoding the event sender endianess
  - Some call related data (Coded using CDR)
    - Receiving event receiver object identifier
    - Method name to be called
  - Event data themselves (Coded using CDR)

- 0MQ multipart messages used to transport these data
The new event system

- 0MQ includes OpenPGM for multicast transport
  - http://code.google.com/p/openpgm

- Spreading the events into multicast group (address)
  - Find a way to automatically distribute the event on the available multicast group

- Unicast is still the default

- Multicast supported for some specific events defined by the system administrator
The new event system

• We are using the 0MQ publisher – subscriber pattern (pub-sub)
  • The device server process is the publisher
  • The applications listening for events are the subscribers

• 0MQ subscription is used to filter out unwanted events
  • Subscription are length-specific blobs
  • Subscriber receives only messages beginning with specified subscription buffer
  • The first string sent in event messages is used as subscription buffer
The new event system

• Implementation
  • We are using 0MQ 3
    • Subscription forwarded to the publisher (Unicast only)
      • Less network bandwidth
      • Less CPU consumption on client side
  • 0MQ is written in C / C++ but it's API is C
    • We are using a provided C++ binding
    • 0MQ also provides a Java binding based on JNI
      • Not ready yet for 0MQ 3

• Compatibility old system – new system
  • A new kind of synchronous request exchanged between event subscriber and publisher during Tango event subscription
Conclusion

• Tango Event system
  • Only client and server processes
  • Better performances than previous system
  • Multicasting requires more attention

• Tango
  • It's still an evolving project
  • Problem is not the lack of ideas but rather a lack of resources
  • We now have a clear way to take decision