TANGO A CORBA BASED CONTROL SYSTEM

A.Götz, E.Taurel, J.L.Pons, P.Verdier, J.M.Chaize, J.Meyer, F.Poncet, G.Heunen, E.Götz
ESRF, 6 rue Jules Horowitz, 38043 Grenoble, FRANCE
A.Buteau, N.Leclercq, M.Ounsy, Synchrotron Soleil,
Saint-Aubin, BP 48, 91192 Gif-sur-Yvette, FRANCE

Abstract

TANGO is a CORBA based control system being developed by the ESRF and Soleil synchrotrons in France [1]. TANGO models CORBA objects right down to the frontend. It offers a complete set of communication services for communicating, logging, events, configuring, persistance as well as a full set of generic graphical clients for setting up and testing a TANGO control system. TANGO has bindings for a number of popular languages and commercial packages. This paper will present TANGO and demonstrate how TANGO can be used to control a Java enabled device, in this case a LEGO [2] Mindstorms robot dancing the tango.

INTRODUCTION

New control devices are appearing on the market with more capabilities than their ancestors. Frequently they are controlled with high speed networks and object oriented software libraries. These devices are a far cry from the old serial line. In addition to the new devices to control new languages like Java have appeared in the recent years which are also object oriented. At the ESRF and Soleil we have built a new control system called TANGO which is object oriented and which is well adapted to controlling these new devices. Our experience with the TACO object oriented control system has enabled us to improve on the past.

TANGO uses CORBA for doing network communication. CORBA is a language independant standard for implementing distributed objects on the network. CORBA is an evolving standard and is continuously adding new features which are of interest for controls e.g. real-time, embedded, component model. TANGO uses the omniORB implementation of CORBA in C++ and JacORB for Java [3]. OmniORB is freely available, open source and one of the fastest ORB implementations around.

TANGO PHILOSOPHY

The philosophy of TANGO is to provide users with an object oriented control system which is powerful and easy to use and which offers the advantages of using CORBA but hides the details. We have adopted the approach that all TANGO objects are derived from one class called Device. Device is the TANGO component. It offers the following features :

- *name* - every device has a name which is unique in every instance of TANGO
- *properties* - device are configured by their properties. Properties are persitantly stored items which describe certain aspects of device configuration e.g. channel address, baudrate, min, max etc.
- *attributes* - each device can have a set of data attributes. Attributes have a fixed description which defines minimum and maximum values, alarm and warning levels, event trigger levels etc. Attributes can be events too.
- *commands* - each device can execute a set of commands, commands accept one input parameter and one output parameter. A parameter can be an array.
- *events* - are asynchronous attributes which are sent to all subscribers.
- *server* - devices are instantiated in a process called a device server. This is a container for a collection of devices. It provides an administration interface to restart the server, read out the black box, reinitialise devices etc.
- *polling* - device commands and attributes can be triggered automatically. This is doing by polling devices inside servers. The polling thread is used to trigger events, fill the data cache, trigger actions.
- *multi-threading* - device servers are multi-threaded. The omniORB implementation of CORBA used by TANGO is thread-safe and allows writing threaded applications using TANGO.

TANGO SERVICES

The TANGO device component uses the TANGO services in order to implement all the features listed above. The following services are offered in TANGO :

- *database* - a system wide database using MySQL is provided for persistant storage of device properties and device names
- *naming* - a naming service is provided via the database which allows devices to be found independant of which host they are running on. Clients and servers from multiple instances of TANGO can connect with each other.

- *event* - TANGO uses the omniNotify implementation of the CORBA Notification service.
- *logging* - a logging service which uses log4j is provided for all devices.
- *archiving* - archiving of historical data is provided.
- *groups* - groups of devices can be constructed on the fly and controlled as a single device.
- *api* - a high-level application programmer's interface is provided in C++ and Java. The api provides stateless connection management.
- *starter* - TANGO servers can be automatically started by a starter daemon.

TANGO TOOLS

TANGO is delivered with a full set of graphical tools for building TANGO device servers, testing them, plotting, logging and archiving. All graphical tools are written in Java using the Swing graphical library. The following graphical tools have been developed so far :

- *pogo* - a graphical tool for generating device classes in C++ or Java. Removes a large part of the tedious work involved and makes device class development very rapid.
- *logviewer* - a graphical tool for viewing and filtering log messages.
- *jive* - a graphical tool for viewing and modifying the database and testing devices.
- *devicetree* - a generic graphical tool for testing devices and building simple monitor panels.
- *atk* - a graphical application toolkit with a large number of graphical viewers for various data types. Can be run standalone or be used to build new graphical applications.
- *astor* - a graphical tool for supervising a TANGO control system e.g. starting device servers, checking device servers are running etc. and testing devices.

TANGO BINDINGS

TANGO has bindings for the following languages :

- C++
- Java
- Python
- Matlab
- Labview
- Igor

DOWNLOAD

The latest version of TANGO (V3.0.2) is available for downloading from the ESRF ftp site [4]. The following downloads are available :

1. **tango-3.0.2.tar.gz** - source code release for Linux and Solaris (built with GNU autotools),
2. **tango-3.0.2.win32.zip** - binary release for Windows.

The TANGO core source code and tools are being developed as a Sourceforge project [5] (**tango-cs**) and the latest versions can be checked out from the CVS repository.

TANGO EVOLUTION

TANGO has already achieved a high degree of sophistication. It is currently being used to build the Soleil synchrotron and to modernise the ESRF control system. In the future TANGO will continue to evolve to profit from improvements to CORBA. The next step is to adopt the CORBA Component Model. This will allow a better mapping between TANGO components and Enterprise Java Beans (EJB). The CCM is very close to the current TANGO device model [6]. The advantage of the CCM is that it offers a common framework with other CORBA systems, more high-level tools and solutions for areas which are currently not addressed in TANGO - security, packaging etc.

Table 1: *List of TANGO device classes written so far*

device class	description
database	database server for MySQL
starter	server to start other servers
scan	generic scanning server
serialline	generic serial line controller
gpib	NI enet gpib controller
paragon	stepper motor controller
mitutoyo	analog encoder
diamond	diamond crystal monochromator
pressure	pressure gauge controller
meteor	framegrabber controller
irmirror	infra-red mirror controller
tunemon	tune monitor controller
wago	generic WAGO controller
wagoadc	WAGO adc controller
wagojacks	WAGO jacks controller
wavemaster	high-end oscilloscope controller
plcmodbus	modbus enabled plc controller
mindstorms	LEGO mindstorms robot controller

LEGO ROBOT

TANGO can be used to interface almost anything. Already a large number of TANGO device servers exist. See table 1 for an (incomplete) list of existing device servers.

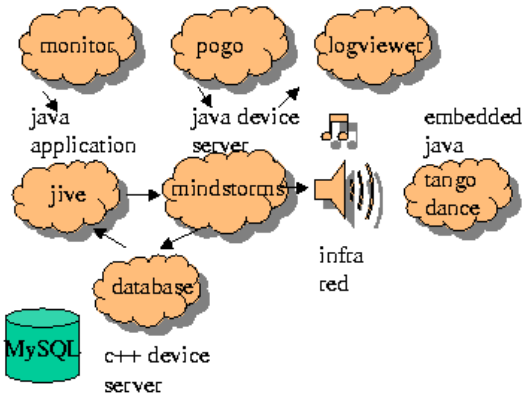

Figure 1: Diagram of TANGO device server architecture for robot dancing tango

In this session we will give a live demo of how to interface TANGO to a LEGO robot dancing the tango.

LEGO brought out a robot kit called Mindstorms in 1998. It consists of a programmable brick with three motors, three sensors and 700 lego pieces. The heart of the LEGO brick is a Hitachi 8300 chip with 32 kbytes of memory. A very innovative programmer (Jose Solarzano) ported a reduced version of the Java virtual machine to the Hitachi. This project was called TinyVM. Today this has forked into the Lejos project [7] which offers a JVM (16 kbytes) as well as a complete set of classes for communicating with the robot from a PC using the IR tower, taking images using the LEGO Logitech camera and classes for doing robotics.

We have used these classes to build an embedded java program which dances the tango and communicates with a TANGO device server written in Java. The embedded java program reads out the light sensors and returns them as an attribute, implements a state machine and implements the following commands :

- DanceTango
- MoveForward
- MoveBackward
- LeanForward
- LeanBackward
- TurnLeft
- TurnRight
- Stop

The communication between the device server and the robot is doing using the java communication classes to send and read integers via infra-red. Because the communication is not very fast (approximately 1 message per second) a thread has been used to implement it so as not to block the

Figure 2: LEGO robot dancing TANGO

TANGO communication. The infra-red communication is the weakest link in the robot and the tower has to be placed quite close to the robot in order to communicate.

CONCLUSION

TANGO is a mature CORBA-based middleware solution for building and distributing control components. CORBA offers the advantages of language independance, efficiency, and events. The next release of TANGO (4.0.0) will include events, and groups. In the future TANGO will integrate the CORBA Component Model. TANGO has been used to interface a number of different hardware already. TANGO can also be used on fun projects like the LEGO robot.

REFERENCES

[1] http://www.esrf.fr/tango - the TANGO website.

[2] LEGO is a registered trademark of The LEGO Group

[3] http://omniorb.sourceforge.net - the OmniORB website.

[4] ftp://ftp.esrf.fr/pub/cs/tango

[5] http://sourceforge.net/projects/tango-cs

[6] http://www.omg.org/technology/documents/formal/components.htm - the Corba Components Model specification.

[7] http://www.lejos.org - the Lejos website.