UBIQUITOUS TANGO

Andy Götz, Jaro Butanowicz, Lukasz Slezak, ESRF, Grenoble (France)

Claudio Scafuri, Giulio Gaio, Elettra, Trieste (Italy)

Abstract

Tango is a control system based on the device server concept. It is currently being actively developed by 5 institutes, 3 of them are new institutes. This alone is a good reason that Tango integrates the latest developments in control systems evolution. One of the evolutions in the computing world is ubiquitous computing. Ubiquitous computing in control systems means integrating computers and intelligence into every aspect of the control system. This paper will present how Tango has been integrated into a wide variety of embedded systems from FPGA's, Gumstix, Libera's and even PS3's (if my boss would buy me one).

INTRODUCTION

This paper discusses how Tango can integrate ubiquitous computing. It discusses how to embed Tango. It presents a number of examples of how Tango has been embedded into ever smaller devices. It also explores the future on how Tango could be used to interface a large number of miniature devices.

UBIQUITOUS COMPUTING

Ubiquitous Computing, also known as the "***Third Wave in Computing***", refers to the fact that computers are becoming more and more pervasive in our environment. Current trends show that in the future computers will be everywhere. They will be integrated into our houses, clothes, environment etc. They will be pervasive and do what they have to do invisibly. Computers will gather information and change our environment based on the input they have and the intelligence embedded in them. This is already partially happening with portable phones. There is a strong likelihood that this trend will be adopted by traditional control systems e.g. accelerator control. This means there will be more and more embedded computers for the control systems of the future to interface to. We see this trend happening already with the multiplication of intelligent devices based on interfaces like Ethernet, I2C, USB, Bluetooth etc.

EMBEDDING TANGO

The Tango control system is based on the concept of device servers. Device servers are processes which run on or close to the hardware. Clients and other servers communicate with the device server via the network using the CORBA protocol. Configuration information and network address are stored in a central database. Tango can run without a database. This means that in its simplest configuration Tango consists of a single device server running on the hardware. The device server uses the hardware's Ethernet communication channel to communicate with clients.

Tango is by its design well adapted to being embedded on the hardware. Here are some of the advantages of Tango for embedding :

- fully distributed
- device server concept is simple
- runs on Linux and Windows
- requires only a C++ compiler and a TCP/IP stack
- can run without central services like database and events
- omniORB CORBA runs on 386, PPC and ARM architectures

These advantages have led Tango being ported to a range of 386, PPC and ARM based architectures running Linux. In fact any processor running Linux, this includes the PS3 CELL processor (if my boss would buy me one!). The procedure to follow when porting Tango to one of these platforms is :

- install a C++ compiler
- compile omniORB for the target
- compile Tango for the target
- write and compile the device server
- download the server and run it on the target

Easy - when you know how! The main difficulties arise from cross-compiling omniORB. On the platforms mentioned this is now well documented. Tango only depends on standard features of the standard C++ library and POSIX threads and usually does not present any problems to compile. We will now present some hardware examples Tango has been embedded on.

LIBERA

Libera is a sophisticated device developed by the company Instrumentation Technologies (Slovenia) for doing beam diagnostics. It is used extensively in many synchrotrons in Europe. The Libera device server was the first example of embedding Tango close to the control hardware. The Libera has an ARM CPU as host. omniORB was ported to the arm architecture for the Libera project. The device server written by Nicolas Leclerq of Soleil was ported by Elettra to the ARM. The resulting solution has the advantage that it makes managing large numbers of Libera's easier. It also reduces network traffic. A full report of embedding Tango on

Libera has been published as part of the PCAPAC 2007 proceedings [3].

FPGA

Figure 1: Tango-on-a-chip

Field Programmable Gate Arrays (FPGA's) are the current trend for a wide range of hardware designs. They offer the advantages of high performance and a high degree of flexibility at low cost. Many manufacturers offer FPGA boards with a "hard core" PowerPC (PPC) processor embedded in the FPGA chip fabric. The CPU has direct access to the FPGA registers. Recently Board Support Packages (BSP's) for Linux have been made available for FPGA boards. This means it is possible to run a Linux application on the PPC embedded inside an FPGA. Another recent development has been the development of Linux drivers for the on board TMAC Ethernet interface. This means it is now possible to run Tango on a hard core PPC on an FPGA chip. Because everything runs on a single chip this project is referred to as ***"Tango-on-a-chip"***.

In the summer of 2007, Jaro Butanowiscz, did a new port of omniORB and Tango to PPC using the latest versions of both packages and the gcc cross compiler. Using the Linux tools provided by Xilinx for their development board ML403, a Linux image with a Tango device server was wrapped together with an FPGA design to run the first FPGA Tango device server with its application on a single chip. The performance of a single network call is 5 ms. The advantage of wrapping the device server with the FPGA application is the reduced number of protocols to manage. The full procedure is documented on the tango web site [1].

GUMSTIX

Gumstix is a series of small computers which are no bigger than a stick of chewing gum (80 mm x 20 mm x 6.3 mm). They are based on an Xscale CPU which is compatible with the ARM. Gumstix provides a ready-to-use Linux with cross development tools (Buildroot). This made it easy to compile omniORB and Tango for the Gumstix platform. Gumstix offers a wide range of interface boards.

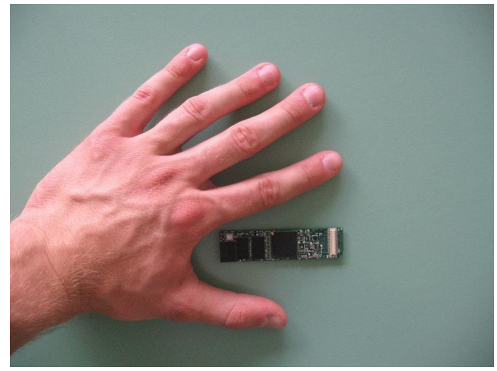

Figure 2: Finger sized Gumstix

In the summer of 2007 Lukasz Slezak evaluated Gumstix with a range of interfaces :

- serial line
- wirefull and wireless Ethernet
- compactflash USB host
- robostix (I2C + ADC + DAC)
- bluetooth

Using the above interfaces Tango device servers were written for the following devices :

- Paragon stepper motor
- Eurotherm thermocouple
- ADS1100 16 bit ADC
- Labjack U3 ADC/DAC
- Wiimote

The Gumstix platform proved to be an easy to use and versatile hardware platform. The Buildroot kit from Gumstix was a big time saver. Gumstix demonstrated the ease with which intelligence and wireless connectivity could be embedded into hardware. The only downside of the Gumstix was the mechanical packaging. Although Gumstix will not be adopted at the ESRF for interfacing hardware this project proved that Gumstix's are great for prototyping ubiquitous computing solutions.

FUTURE HARDWARE

Ubiquitous computing means we can expect to find lots of very tiny computers to integrate into our control system. A typical example of such hardware are a new generation of tiny sensors called motes.

"***Motes or Smartdust*** *are a hypothetical network of tiny wireless micro-electromechanical systems (MEMS) sensors, robots, or devices, installed with wireless communications, that can detect (for example) light, temperature, or vibration."*[1]

[1] Wikipedia : http://en.wikipedia.org/wiki/Smartdust

Figure 3: Pen point sized mote

Despite their small size, motes have a CPU, ADC and wireless network on board. Interfacing thousands of devices like this represents new challenges. An open source operating system called TinyOS is being developed at Berkeley which would allow motes to communicate with classical operating systems like Linux. One could imagine a wireless Gumstix running a Tango device server used as a concentrator for a few tens of up to a hundred wireless motes running TinyOS.

MORE “UBIQUITOUSNESS”

Ubiquitous computing means interfacing large numbers of wireless computers in a transparent manner. How to improve Tango to be even more ubiquitous? Here are some areas of research for the future :

footprint – reduce the footprint of Tango (currently 5MB) for embedding into even smaller systems

zeroconf – integrate support for zeroconf to dynamically allocate the IP address and dynamically export the device(s)

private networks – due to the lack of Ethernet connections we need to create private local networks and access servers on the private network from the public network

multicast – add support for multicast protocols to increase client bandwidth

dynamic discovery – add support for dynamic discovery of servers by clients i.e. without going via the database

gis – add geographic information to each device as a standard attribute in order to find them

Ubiquitous Tango means not only integrating large amounts of hardware. It could also means large numbers of analysis codes i.e. the grid. In the future Tango could be embedded into analysis programs to do online data analysis.

CONCLUSION

Tango is being embedded into increasingly smaller devices and is ready for the next wave in computing - ***Ubiquitous Computing***. Tango could be made even more ubiquitous by developing the topics identified in this paper. In the future Tango could be everywhere !

REFERENCES

[1] TANGO home page : www.tango-controls.org

[2] Mark Weiser's UbiComp page : http://www.ubiq.com/hypertext/weiser/UbiHome.html

[3] G. Gaio “Embedding Tango on a digital BPM”, PcaPAC 2006 (Newport, USA)