TANGO, TACO

- TANGO is TACO's son (or daughter)
  - Object oriented
  - Device server
  - Unified device access
- TANGO uses more modern technologies

	TACO	TANGO
Database	ndbm/oracle	MySQL
Communication	RPC	CORBA
Language	C	C++/Java
Platform	Linux/HP/Solaris/NT/OS-9	Linux/HP/Solaris/NT
Programming style	Object in C	Object oriented

CORBA

- CORBA is a communication layer between objects
- TANGO uses CORBA but
  - Hides it to the programmer
  - Does not uses CORBA services
- ORBacus from OOC
  - Future ???
- One common IDL file instead of one IDL file per device class
  - Allow writing of generic application
  - Keep system simple

TANGO device

- Each object to be controlled
- Has a unique name
  - [protocol://facility/]**domain/family/member**[/attribute.property][?dbase=yes]
    - sr/d-ct/1
    - Tango://gizmo:20000/sr/d-ct/1?dbase=yes
    - Taco://margaux/id12/pen/11
    - myhost:2345/id10/rv/1?dbase=no
- Each device belongs to a class
- Device supports
  - A state
  - Commands
  - Attributes
  - CORBA attributes
  - Miscellaneous utilities

TANGO device

- Device state
  - 14 pre-defined states (ON, OFF, FAULT, STANDBY, UNKNOWN….)
- Commands
  - A generic call
    - out = dev.command_inout(“Cmd name”,in);
    - out = dev.command_inout(“Cmd name”,in,source);
  - 20 different data types
  - Each device automatically has two commands DevState and DevStatus
  - Query device command list
    - dev.command_list_query()

TANGO device

- Attributes
  - Command transport data which are not normalized. No info on data meaning. Generic display program cannot interpret data.
  - Attributes = data + additional information like min, max, unit, desc….
  - Attributes are zero,one or two dimensions data.
  - Only four data types (short, int, double, string)
  - Four calls to deal with attributes
    - read_attributes() from device or from cache
    - write_attributes()
    - get_attribute_config()
    - set_attribute_config()

TANGO device

- Standard attributes
  - Five standard attributes supported :
    - state, status, name, description and administrator device name
- Miscellaneous calls
  - ping()
  - info()
  - black-box()

TANGO device pattern

- A common framework for all control programmers
- Writing control software for a new kind of devices means writing a new implementation of this device pattern
- Based on commonly used object oriented design pattern (singleton, command)
- Two set of classes
  - Base classes merged in a library
  - User classes with the hardware access

TANGO device server

- An operating system process with one or several device pattern implementation
- Automatically contains one implementation of the device pattern for the DServer class implementing administration commands
  - Polling thread configuration/status
  - Changing verbose level
  - Killing process
  - Restarting device/device pattern
- Uniqueness achieved using an instance name during startup
- Configurable via database or via command line for non database device server

TANGO database

- The database is used to store
  - device and class properties (device configuration parameters e.g. board address, initial values..)
  - attribute properties (class and device)
  - device network address and names (IOR)
- The database is accessed like any other device
- TANGO uses MySQL as database engine
- The database device server is started on a known host and port.

TANGO API's

- Two API's supported to write TANGO clients (Java and C++)
- The API rules are
  - Hide two steps connection mechanism
  - Hide reconnection in case of front-end computer reboot or server shutdown
  - Manage IDL versioning
  - Gives an easy access to all database features
  - Hides the remaining CORBA details

Platforms/Performance

- TANGO is supported on four platforms
  - Linux (Suse), Solaris, HP-UX (10.20)
  - Windows (98 - NT)
    - In a MS-DOS window
    - As a classical windows application (Win32 and MFC)
    - As a service (NT)
- Performance is not a bottleneck

From - To	Network	Time
Linux - Linux	Same computer	0.5 mS
Linux - Linux	100baseT	0.7 mS
Solaris - Solaris	Same computer	0.8 mS
Solaris - Linux	100baseT	1 mS
WinNT - WinNT	10 baseT	0.9 mS

TACO/TANGO compatibility

- TACO to TANGO
  - TACO API modified to deal with TANGO device for compatible feature. Switch based on device name
- TANGO to TACO
  - C++ : Use classic TACO calls
  - Java : A JNI layer between a JAVA application and the TACO library.. Switch based on device name

TANGO tools

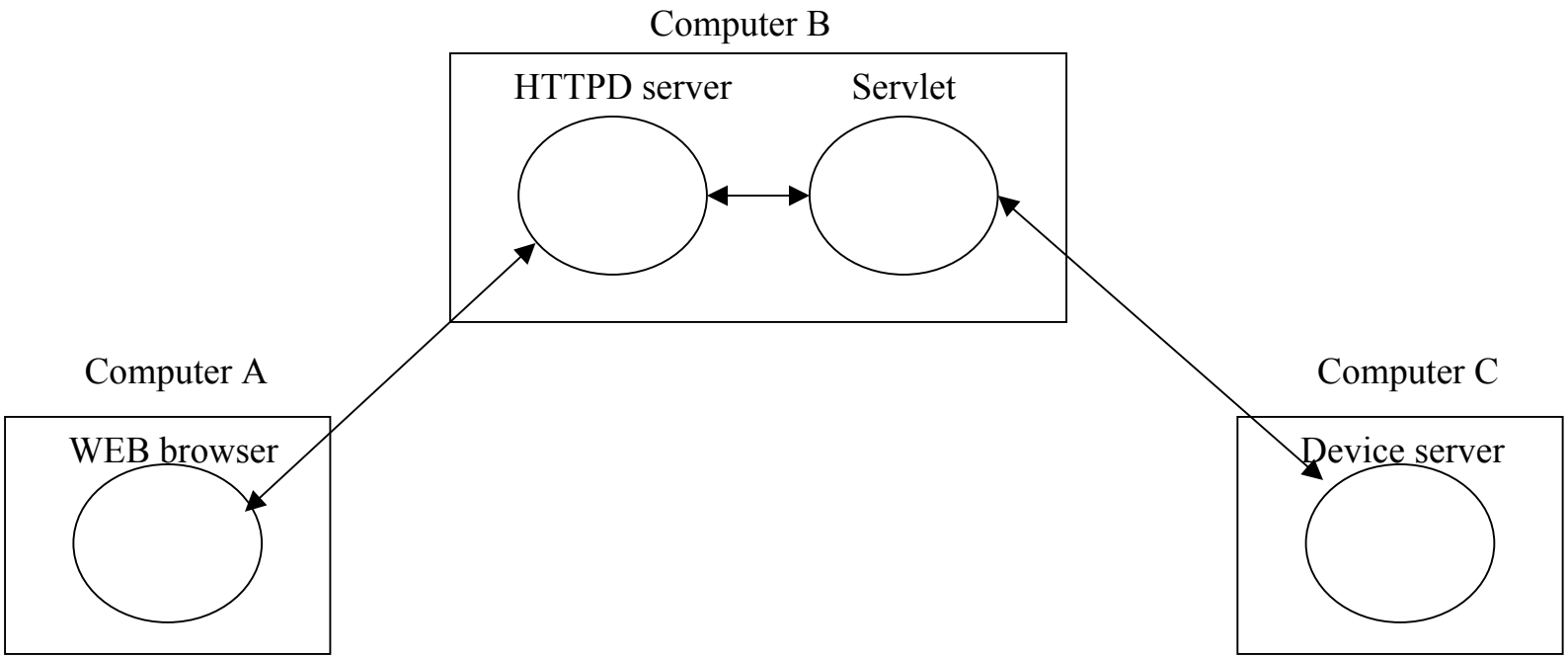
- Code generator (Pogo)
- Control system administration tool (Astor)
- WEB generic device client and database interface (Jive)
- LabView interface

POGO : A code generator

- POGO is a source code generator for TANGO device server
  - Able to take its inputs from an already written source code
  - Generate doc from source code (HTML format) if you follow some conventions (Javadoc like)
  - Help you to port TACO device server to TANGO device server
  - GUI using Swing/JDK 1.2

Jive : A generic WEB interface

- Two usages :
  - Generic device menu
  - Graphical TANGO database interface



Astor : Administrating TANGO

- Ease startup/shutdown of :
  - Device server using a specific *Starter* device server
  - Tango system
- Check devices
- Sequencing of startup/shutdown of TANGO control system
- Written using Java/Swing

Future

- Asynchronism/Event --> In progress
- Complete Jive (db/dev interface) --> In progress
- Test/Port to a new ORB
- Packaging --> In progress
- New DB features
  - property history
  - tag
- Security
- Application framework --> In progress
- Link TANGO-HDB
- Doc (One book)
- Python interface (Client and server)