# The State of TACO in 2001

- Mission
- Humble Beginnings
- Features Chronology
- State of the Art
- Future

# The TACO Mission

- Vision
  - a control system is a sea of networked objects ; every part of the control system should be a network object whether it is a low-level, high-level or application-oriented function
- Mission
  - to provide the best software for building networked objects and make it easy to use and freely available

# TACO - Humble beginnings

- 1990 - a simple diagram, a simple api
- 1991 - first server controls a single bit
- 1992 - device servers control linac +booster
- 1993 - storage ring + first beamline

# TACO - Features Chronology 1

| | | |
|---|---|---|
| 1990 | Research<br>Chose ONC/RPC | Version 0 |
| 1991 | Synchronous device servers<br>Specific XDR data types | Version 1 |
| 1992 | Source control under RCS<br>Added generic XDR data types | Version 2 |
| 1993 | Put commands + errors in database<br>Defined Dserver Teams | Version 3 |
| 1994 | Security<br>Support for in-process clients | Version 4 |
| 1995 | Servers in C++<br>Tcl/Tk clients<br>Ported to Linux | Version 5 |

# TACO - Features Chronology 2

| | | |
|---|---|---|
| 1996 | Added support for multiple nethost<br>Added signal data types | Version 5 |
| 1997 | ASynchronous device servers<br>Ported to VxWorks<br>Matlab clients | Version 6 |
| 1998 | Java api<br>Source code release + GPLed | |
| 1999 | Added events<br>Python clients<br>Labview clients + servers | Version 7 |
| 2000 | Added support for TANGO<br>Dynamic error handling | Version 8 |
| 2001 | Workshop | |

# TACO - Device Model

- Commands
- State machine
- Client-server
- API
- Database
- Manager
- Security
- Signals

# TACO - Platforms

- Linux on x86 + 68k
  - add ARM + PowerPC + 64bit
- Windows 95/98/NT/2000
  - updated November 2001 (V8.30)
  - update C++ port
- Solaris on SPARC
- OS9 on 68k
- HP-UX on HPPA
  - will die out ?
- Abandoned
  - VxWorks + LynxOS + Irix

# TACO - Server Languages

- C++
  - workhorse
- C
  - ideal for embedded (footprint=25 kbytes)
- Python
  - cannot be simpler
- Labview
  - use as shared library

# TACO - Client Languages

- C
  - workhorse
- Python
  - scripting language of the future
- Labview
  - quick (and dirty ?)
- Matlab
  - useful for scientific applications
- Tcl
  - will it die out ?

# TACO - Communication

- synchronous
  - improve reimport algorithm
  - make TCP default protocol
- asynchronous
  - used quite a lot
- event-driven
  - not used at the ESRF !
  - re-subscribe to event if server restarted
- multiple control systems

# TACO -Performance

| | | |
|---|---|---|
| Linux on a dual Pentium III 930 MHz | in same process | 5 microseconds |
| Linux on a dual Pentium III 930 MHz | process to process on same computer | 60 microseconds |
| Linux on a dual Pentium III 930 MHz | process to process on different computers | 420 microseconds |
| Solaris on a SPARC | process to process on same computer | 240 microseconds |

# TACO - Footprint

| | |
|---|---|
| C server (shared) | 25 kilobytes |
| C++ server (shared) | 250 kilobytes |

# TACO - Threading

- single server thread for network
- server can setup polling thread
- client calls can be threaded
- objects can be in-process or out-of-process
- multiple worker threads supported
- when in doubt use mutexes

# TACO - Database

- ndbm
  - replace with MySQL
- Oracle
- greta
- dbase tool (FRMII)

# TACO - Data caching + Archiving

- Data collector
  - Make it simpler ?
- Archiving database (HDB)
  - Port to MySQL
  - should this be exported ?
  - Make it easier to configure

# TACO - Applications

- Automatic class generator (is rarely used anymore)
- Fsigmon displays signals live
- Staticmon displays signals from archive database
- Xdevmenu for testing any device
- Specific applications for machine control
- SPEC for beamlines

# TACO - Packaging

- Source code release via ftp
  - redo using autotools
- External access to source code tree
  - *taco.sourceforge.net* already registered
  - propose to move TACO source code there
- Documentation
  - 270+ pages of TACO manual
  - web site needs improvement

# Worst of TACO

- Our worst errors were
  - Not implementing events earlier
  - Defining two api's (ds + dc)
  - Allowing specific types
- Worst platform was OS9
- Worst database was ndbm
- Worst algorithm was the reimport one (20 second wait)
- Main missing features are
  - clean startup procedure
  - good administration tool
  - alarms
  - SQL database
  - C++ api
  - Web support

# TACO and TANGO

- TANGO is being developed using modern tools
- TANGO will improve TACO and avoid the same mistakes
- However TACO will continue to survive (ESRF has a huge TACO installation)
- TACO is complementary to TANGO (simpler and lighter BUT fewer features)
- PROPOSAL :
  - Make TACO and TANGO 100% compatible
  - TACO clients can call TANGO clients and vice versa in C++ and Java (switching will be done by API's)
  - TANGO servers can use TACO classes and vice versa in C++ (develop wrapper classes to do this)

# TACO - The Future

- TANGO is the future @ ESRF
- full interoperability between TACO + TANGO
- minor improvements to TACO (adapt or die)
- provide public access to code tree
- GOALS
  - be as good if not better than the commercial solutions (e.g. OPC, commercial PLC's) while being easier to use
  - write device servers for ALL parts of the system
  - implement the killer idea, device server and/or client