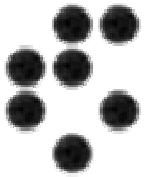


Jožef Stefan Institute

cosylab
CONTROL SYSTEM LABORATORY

The Babylonization of Control Systems

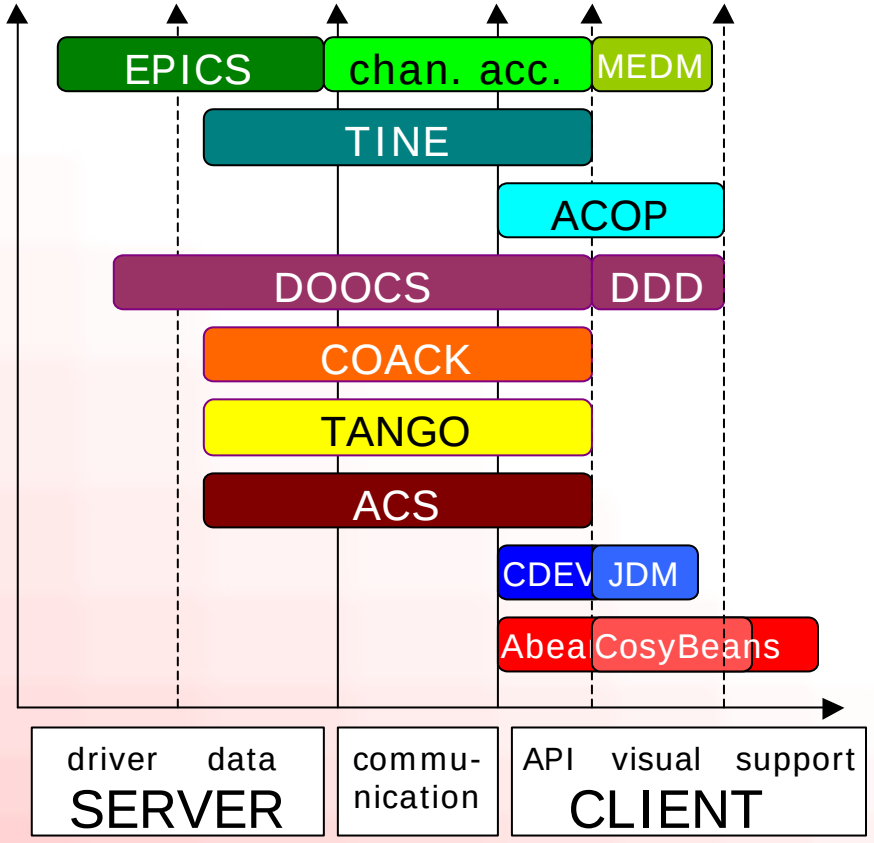
P.Duval, Z. Kakucs - DESY

M. Kadunc, I.Kriznar, M. Plesko, A. Pucelj, G. Tkacik – JSI and Cosylab

Available Control Systems

- Complete systems
  - ACS
  - COACK
  - DOOCS
  - EPICS
  - TANGO
  - TINE
  - ...
- APIs and GUIs
  - Abeans and CosyBeans
  - ACOP
  - CDEV
- Machine physics packages
  - Databush
  - XAL

The Layers of Control System Packages



CS Comparison: Basic Features

	EPICS/CA	TINE	DOOCS	COACK	ACS	TANGO	TACO
monitors	on-change	on-change, on-timer, peer-to-peer, peer-to-network	As per TINE or CA	On-change On-timer interval	on-change, on-timer, threshold and interval settable for each client	on change, on alarm, periodic	server specific
groups of signals	Partially (buffers requests)	YES	YES via TINE	YES (*command)	NO	future	yes via data collector
built-in simulation	YES	YES	YES	YES (Virtual-machine)	external	NO	NO
Dynamic addition of changes	Changes dynamic,Add d/Remove needs reboot	Minimal. Security changes (server), address changes (client) do not require reboot.	Device Address changes do not require reboot	Comonent base addition does not require reboot	add/remove device	YES via database	YES via database

CS Comparison: I/O Integration

	EPICS/CA	TINE	DOOCS	COACK	ACS	TANGO	TACO
Config Database	text-based .db file, templates + substitutions	Distributed (unspecified)	Distributed (DOOCS)	XML based	XML-based, hierarchical data supported	MySQL	ndbm, Oracle or MySQL
Eplicitly supported I/O platforms (e.g. VME, PCI, CAN-bus,…)	VME, cPCI (in test), CAN, GPIB, VXI, Allen Bradley, Serial, PLCs, BitBus, and others…	SEDAC	SEDAC, PROFIbus, CANbus, Shared MEM		CAN-bus, LonWorks	OPC	serial line, gpib, modbus
Explicitly Supported I/O cards	see: www.aps.anl.gov/epics/modules/bus.php	n.a.	All with standard UNIX driver	Anyone with Win32 drivers	n.a.	growing list	large list
support for external timing	YES	NO (do-it-yourself)	YES	NO (do-it-yourself)	YES	NO	NO

CS Comparison: Services and Tools

	EPICS/CA	TINE	DOOCS	COACK	ACS	TANGO	TACO
Event service (data channel)	via CA gateways	YES (NETMEX)	Via TINE NETMEX	YES	YES	work in progress	yes
Alarm services	Distributed	Distributed + Central	Distributed + Central	YES	Distributed	log service	no
archive services	Central + Distributed	Distributed + Central	Distributed	YES (in Database)	central, through notify service	via logging	YES (uses Oracle)
postmortem services	NONE	YES, Event driven	YES		NO	black box	NO
Alarm Viewer	YES	YES	YES	YES	from Abeans		
Log Viewer	CMlog	YES	YES		YES	Chainsaw	
Sequencer	FSQT, SNL	NOT Standard, but in place for HERA	YES		NO	Python	Python, SPEC or Tc

CS Comparison: Application Development

	EPICS/CA	TINE	DOOCS	COACK	ACS	TANGO	TACO
API	Active-X, Python, Perl, Fortran, IDL, Java, tcl, C, C++	ActiveX, VB, C, C++, Java, MatLab,	C, C++, FORTRAN, Java, MatLab, LabView	COACK components for VB/LabView /HP-VEE	device based through CORBA IDL (BACI model)	C++, Java, Matlab, Labview and Python	C, C++, Python, SPEC, Tcl, Labview, Matlab, IDL
GUI builders	DM2K, MEDM, EDM	VB, JAVA IDEs + TINE client wizard for VB and Java, DDD, LabView, HPVEE	DDD, LabView, MatLab	VB LabView HP-VEE	uses Abeans	Forte, QT-designer, Matlab, Labview	BuilderAccessory, QT-designer, Labview, Matlab
GUI libraries	from EDM	ACOP, Cosybeans, VB, MatLab, ActiveX, LabView	LabView, MatLab	VB LabView HP-VEE	CosyBeans	ATK	NO
scripting tools	tcl/tk, Perl, Python, UNIX	Command line calls can be used in standard UNIX scripts	Standard UNIX		Python	Python	Python, SPEC, Tcl

The Really Difficult Questions

- Applications developers:
  - How easy is it to learn the system?
  - How easy is it to create an application?
  - What features are missing?
  - What features are good?
  - What are the limitations?
  - Are the applications primitive (connecting data to display tools only)?
  - Can the applications be programmed?
- Operators:
  - How good are the applications in general?
  - What's missing?
  - What could be better?
  - What is the information density on a typical application?
  - How quickly can I find and isolate problems?
  - How often do failures in the control system hinder operations?

Other Difficult Questions

- Engineers
  - How easy is it to remotely test my equipment?
  - How easy is it to find problems?
  - How good is the alarm system notification of pending hardware failures?
- Inventors of the CS
  - what's your problem? Everything is perfect and there is nothing better!
- Machine physicists
  - How good are the on-line analysis tools?
  - How easy is it to get the live data (archived data) into my favorite analysis tool?
- Directors of institutes
  - Could machine operations be better/smoother/less expensive with another system?
  - mostly care about PR ✍

Mixing Orange Clients with Apple Servers

1. ‘Apple-to-orange’ gateway
   a separate process utilizing the client/server APIs of both systems
2. Client-side ‘apple-plugs’
   client program developers think they are talking to oranges, but they are really speaking native ‘apple’
3. Server-side ‘orange-translators
   server IOCs think they are being addressed by apples but are really speaking native orange.

Why Phil Thinks Translators are Better

- Server-side systematics are guaranteed to be there.
  - local alarm server, local history server, queries, etc.
- The data are as close to the source as possible
- can use the best applications for each problem
- Gateways bring a host of intermediate problems
  - e.g. connectivity problems are difficult to locate
  - another link in the chain

Why Mark Agrees with Him … …but Is Still Happy with Abeans

- Client-side plugs usually provide only
  - common functionality over several CS
    - if the server-side features are not available, you come up empty
  - generic API
    - requires knowledge of CS and is therefore practically useless
- Abeans have a solution though
  - allow different CS data models (Channels, Devices, etc.)
  - interfaces are facades for services (plug-ins) with always at least one default implementation
  - we work hard to provide each single feature in a generic way – if TINE has ArchiveReader, also EPICS gets it!

EPICS, TINE and DOOCS Translator

- run EPICS2TINE directly on the IOC
  - don't speak channel access at all and access EPICS database directly
  - we have also elegantly solved the 16 Kbyte/message barrier (i.e. 4000 floats) of the old EPICS release
- DOOCS servers and clients offer the traditional SUN RPC interface as well as a TINE interface
  - EPICS IOCs are immediately available to DOOCS DDD clients. Using TINE2EPICS, the DOOCS IOCs are likewise available to EPICS MEDM clients.
- Likewise, running Abeans with a TINE plug will see all IOCs as TINE servers irrespective of their parentage.

Abeans plugs for TINE and EPICS

- DESY: Abeans-TINE plug
  - run TINE client applications on non-Windows machines
  - keep access to the full TINE API and services
  - Abeans provide this, plus add a rich client framework
- SNS: Abeans-EPICS plug
  - already has XAL: API to EPICS + machine physics package
  - now wants Abeans as a layer between XAL and EPICS, because of its many advanced capabilities
- Design guidelines:
  - full encapsulation, while keeping all functionality
  - generic solutions preferred
  - use standard Java APIs where possible
  - code decoupling

A Future Scenario?

- Take the “best of all possible worlds”
  - EPICS and DOOCS to integrate VME I/O cards (because they have the drivers)
  - TINE as the access protocol (for mulitcast capability)
  - DOOCS DDD or COACK (for developing synoptic GUI panel)
  - ABeans/CosyBeans (for complex applications and multiplatform capabilities)
- Still use all existing conventional applications
- Further possibilities include advanced features
  - TINE archiver
  - ACS logging
  - Abeans resource loading, etc.

Conclusions

- With translators and plugs we can mix any set of apples and oranges
- Appeal to CS package developers
  - don't reinvent the CS core over and over
  - think more about services that could be used in a generic way by other control systems
  - provide good general purpose applications and tools
  - good documentation and tutorials won't hurt either
  - this is where we are the weakest
- The Tower of Babel fell because of bad monocultural applications!