



Common Templates and Organisation for Beam Instrumentation Front End Software Upgrade

Stephen Jackson

CERN SL / BI Software

16th October – PCaPAC 2002 Frascati, Italy



Who am I?

- ✍ **Computer scientist from England**
- ✍ **First arrived at CERN in 1995**
- ✍ **Involved in many different projects for beam instrumentation**
 - ✍ **SPS (Orbit, Trajectory)**
 - ✍ **LEP (Bunch measurement :Cameras, Telescopes, etc)**
 - ✍ **Transfer Lines (Trajectory from PS->SPS->LEP)**
- ✍ **Developer of control & data extraction software for beam instrumentation**



A bit of history...

- ✍ **CERN relies heavily on software**
- ✍ **Many instrumentation systems created**
 - ✍ **By permanent & *temporary* personnel such as Fellows, PhD students etc**
 - ✍ **Using languages such as Fortran, Modula 2, C, C++ etc**
- ✍ **CERN is a research environment**
 - ✍ **Resulting in *state-of-the-art* systems**
 - ✍ **But...**
 - ✍ **When personnel leave, their expertise leaves too**
 - ✍ **... leaving the rest of the team to pick up the pieces!**



LHC on the horizon...

- ✍ With the advent of LHC, many systems need (re) development
- ✍ Perfect opportunity to standardise before the production of 50+ new systems
- ✍ The BISCO TO project was born
 - ✍ **Beam Instrumentation Software Common Tools & Organisation**

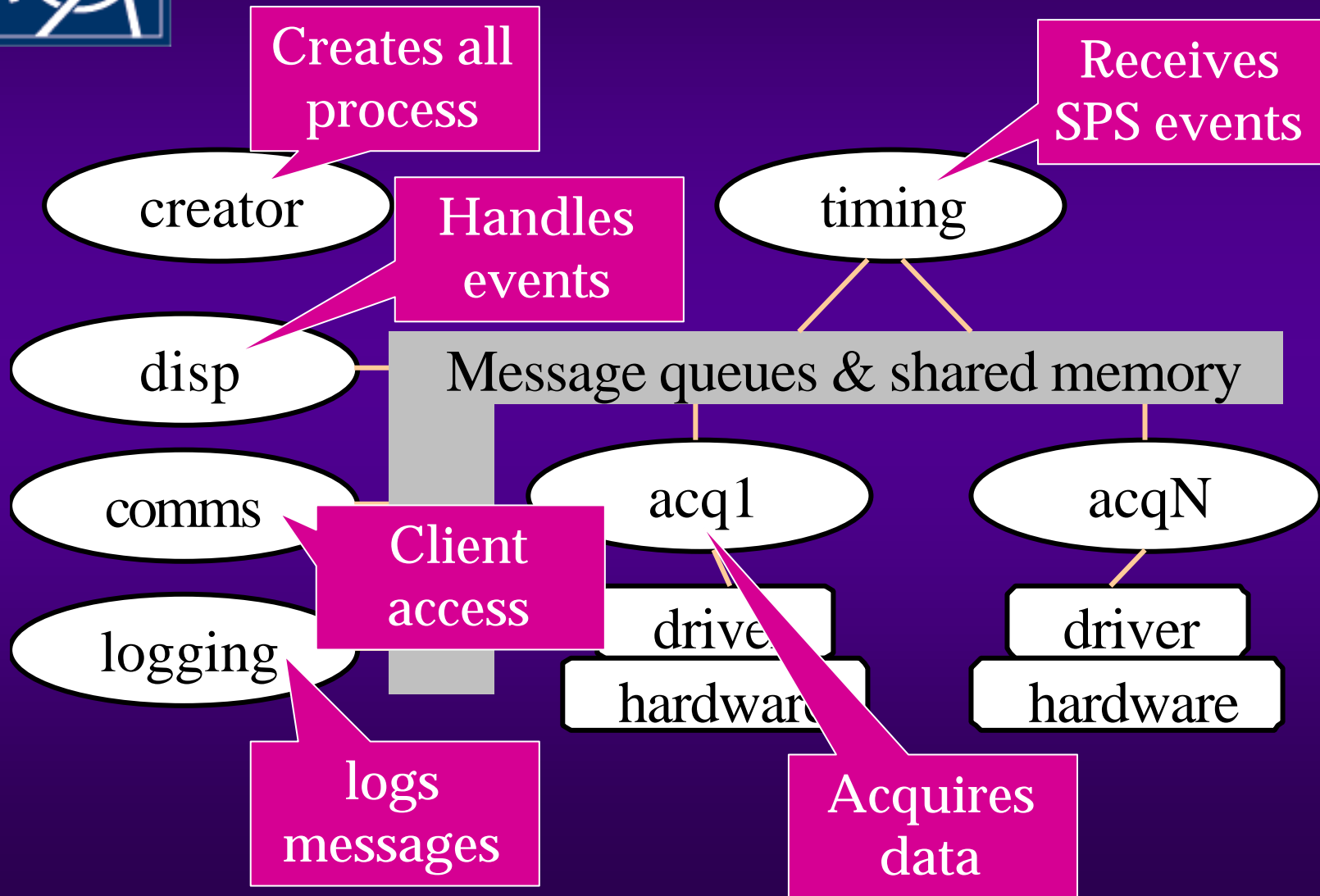


BISCoTO's aim...

- ✍ **Combine the skills of the development team to create a standard real-time template**
 - ✍ **Capable of satisfying all the needs of the team**
 - ✍ **Avoiding instrument specifics – no redundancy**
 - ✍ **Data driven so minimal changes required for a system**
- ✍ **Create generic tools for use by ALL BISCoTO developers / users**
 - ✍ **Only possible if data driven**



BISCoTO's real-time





Data driven

- ✍ **All processes are data driven to avoid divergence from the template**
 - ✍ **Instrument specific code 'include'd at defined places**
- ✍ **Definition name-space**
 - ✍ **Class.Process.Definition**
 - ✍ **Many definitions for many processes for 1 Class of instrument**
- ✍ **Data name-space**
 - ✍ **Class.System.Qualifier.Process.Definition**
 - ✍ **A class of instrument has many systems, ordered by qualifiers**



The Config tool

Class.Process.Definition

Definition entry

CLASS PROCESS DEFINITION MAX_INSTANCES DESCRIPTION VERSION

Field Idx	Field Comment	Field Type	Field ID	Size/Value	Viewable by	Editable by	New group
1	...	CHAR_A	SWExp
2	...	INT	SWExp
3	...	INT	acqCommand	1	SWExp
4	Bisc: Counter incremented by u...	INT	refreshCount	1	SWExp
5	Bisc: Last 'critical' change time (...)	CHAR_ARRAY	refreshTime	80	SWExp	SWExp	ALL
6	Bisc: Event Source (0 -> UsrCm...	INT	eventSource	1	SWExp	SWExp	ALL
7	Bisc: Cycle Detailed Description.	CHAR_ARRAY	cycleDesc	80	SWExp	SWExp	ALL
8	Bisc: Human Cycle Name (Coul...	CHAR_ARRAY	cycleName	32	SWExp	SWExp	ALL
9	Bisc: Human Sequence Name (...)	CHAR_ARRAY	mtgSequenceName	32	SWExp	SWExp	ALL
10	Bisc: MTG cycle	INT	mtgCycleId	1	SWExp	SWExp	ALL
11	Bisc: Cycle Type	INT	cycleType	1	SWExp	SWExp	ALL
12	Bisc: Cycle Num	INT	cycleNumber	1	SWExp	SWExp	ALL
13	Bisc: Will allow to have a set of ...	INT	usrCmdId	1	SWExp	SWExp	ALL
14	Bisc: Will allow to have a set of ...	INT	bstCmdId	1	SWExp	SWExp	ALL
15	Bisc: Automatically set by the ac...	INT	acqMemOffset	1	SWExp	SWExp	ALL
16	Bisc: Depth requested for this c...	INT	acqMemSizeReq	1	SWExp	SWExp	ALL
17	Bisc: Depth obtained.	INT	acqMemSize	1	SWExp	SWExp	ALL
18	Bisc: State of the Cycle/Cmd (1 -...	INT	cycleState	1	SWExp	SWExp	ALL
19	Bisc: Requested state for this cy...	INT	cycleReqState	1	SWExp	SWExp	ALL

Comment

Field ID

Field exposure

Field Type

Optional Size/Value



Mapping C to Java

- ✍ **Server written in C, Clients written in Java**
 - ✍ **Different data types, so BISCOTo data types limited**
 - ✍ No unsigned types for example
 - ✍ **C structs don't map to Java classes under current middleware**
 - ✍ May change with Corba's data marshalling... in 1-2 years
- ✍ **Config tool automatically creates C code & Java Classes for handling data**
 - ✍ **Generated Java class translates C struct into Java bean**
 - ✍ **Generated C code allows anonymous manipulation of server data in files and memory with run-time 'where clauses'**
 - ✍ **All code regenerated automatically when a definition changes**



Generic data input tool

Data file selector

The screenshot shows a software interface with a file tree on the left and a data entry table on the right. The file tree includes folders like BICONFIG, BIFGEN, SHARE, BISCC, DRIVES, TES, NAVIGATOR, MOPOS, BISKEL, IBMS, EXPERT, SCINT, TTPOS, BTVSPS, TOOLS, WSCAN, SCINX2, LOPSYS, BISMAT, and CTLDWC. The data entry table has columns for instanceNa..., instanceInd..., acqConfIndex, refreshCou..., refreshTim..., eventSource, cycleDesc, cycleName, mtgSequen..., and mt. The table contains three rows of data.

instanceNa...	instanceInd...	acqConfIndex	refreshCou...	refreshTim...	eventSource	cycleDesc	cycleName	mtgSequen...	mt
10	0	0	0	No 'Critical'...	1	No Descrip...	Proton FT 1	mtgtab_946	0
10	0	1	0	No 'Critical'...	1	No Descrip...	Proton MD	mtgtab_946	0
10	0	2	0	No 'Critical'...	0	No Descrip...	UsrCmd G...	None	0

Data input table.
Data integrity validation
based on definition



Generic Debugging

- ✍ Logging information is vital when a system has problems
- ✍ BISCO TO uses a separate logging process to handle this task
 - ✍ Performs the time-consuming task of file writing
 - ✍ The real-time tasks simply request a log
 - ✍ Logging is fast (<1ms) and reliable
- ✍ Logs viewed through another generic GUI
 - ✍ Allowing viewing and remote control of logs...



Logging GUI

Select host to interrogate

The screenshot shows the 'Log Control' window. At the top, there is an 'Exit' button and a 'Host:' dropdown menu currently set to 'bmu10t'. To the right are 'Refresh' and three radio buttons for 'Disabled' (selected), 'Continuous', and 'Periodic', followed by a time interval input field set to '0' seconds. Below this is a list of 'Available Logs' on the left, with 'TTPOS/acq_10' selected. The 'Log Details' panel on the right shows 'Name: TTPOS/acq_10', 'Filename: /user/biswop/logging/TTPOS/bmu10t/acq_10', 'Maximum File Size: 500', and 'Verbosity: BISW_LOG_NONFATAL'. There are also checkboxes for 'Mode' with 'BISW_LOG_PROCESS' checked. At the bottom, there are tabs for 'Common Log File', 'Log File', 'TG8_1 Event History', and 'TG8_2 Event History'. The 'Log File' tab is active, displaying a log stream with entries like '11/10/2002 16:41:33 _NF STATS -> acqAcquireReseIFIFO took 0.000109 (0.119997)'. A 'Filter:' dropdown is set to 'BISW_LOG_STATUS' and a 'Grep String:' input field is empty.

Select an available log

Control the logger

The log



Server exploration

- ✍ Developers and users need to interrogate a BISCoTO server's properties
 - ✍ Server template is pre-programmed to allow 'definition' exploration
- ✍ The Navigator can *explore* definitions within a BISCoTO server given its location
 - ✍ All definitions made in the configuration GUI available
 - ✍ Uses the automatically generated Java beans
 - ✍ Involves NO effort from the developer – free!



The generic Navigator

Select the definition in explored tree

Double clicking a definition loads the Java Bean's editor

The screenshot shows the generic Navigator interface. On the left is a 'Selection tree' with a folder structure: CTL, TTP, 2_bmu02t, and BIPEEK. Under BIPEEK, several files are listed, with 'TTPOS.Shared.AcqConf' selected. The main area is an 'Editor' window with a URL bar and a table of fields. The table has columns for 'Field I', values, and 'Action'. The 'Action' column contains 'Get' and 'Set' buttons for each field. At the bottom, there are buttons for 'Get ALL', 'Excel', 'File import', and 'File export'. The status bar at the bottom shows 'Navigator User Level HWEXP', 'Group -> ALL', and 'Actions -> ACTIONS'.

Field I		Action
ActionReqState		Get Set
FrDelay		Get Set
ResetWidth		Get Set
CalibWidth	5.0	Get Set
IntegratorWidth	1000.0	Get Set
CalibConvDelay	9.5	Get Set
ConvDelay	9.5	Get Set
ConvWidth	1.0	Get Set
Page	PAGE0	Get Set
Freq200Mhz22Mhz	22MHZ	Get Set
PosSingle	POS	Get Set
Filter	OFF	Get Set

Fields can be Get and Set



Navigator also triggers actions...

Parameter panel built based on definition

The screenshot shows a web browser window with the URL: mequip://TTPOS.2_bmu02t/BIPEEK?uo=TTPOS.Shared.Ac... and a BISCO TO logo. Below the URL bar is a 'Compulsory filter' section with a 'property' field containing a question mark and a '(string)' label. Underneath is a 'Data Expected in GET' section with a table:

Exp	int
Ex	1

A context menu is open over the 'LoadSave' action, listing the following actions and their available methods:

- ReadWriteString
- LoadSave (highlighted) with sub-menu: GET, SET, HELP
- CycleList
- InitHwStatus
- InitHw
- UpdateSequence
- KillServer
- SendUserCommand
- Reconfigure
- StartStopAcq

At the bottom of the interface, there is an 'ACTIONS' button and a 'GET LoadSave' button highlighted in yellow.

Clicking here gets available actions

Actions can be GET or SET



Other advantages

- ✍ **The tools are generic so...**
 - ✍ **Enhancements made to tools available to all systems**
 - ✍ **Heavy use of the tools, means bugs eradicated quickly**
- ✍ **Automatic documentation of systems**
 - ✍ **API for all BISCO TO systems is automatically documented in a web page using comment, type, exposure etc.**
- ✍ **A lot of data is shared across all systems**
 - ✍ **So some configuration can be centrally maintained**



A BISCoto system for instrument XYZ

Pre-built
Real-Time

Template
modified
with XYZ
as Class

Generic
BISCoto
processes

Free Tools

Definition
Editor

Included
User Code

Config
Editor

Developer
Input

BISCoto
Definitions

LogViewer

User Code
For Drivers

Navigator

Generated
Code

Code for h/w
access based
on definition

XYZ C lib

XYZ Beans



Conclusion

- ✍ **BISCoTO aimed to create a standardised template, maintainable by all...**
- ✍ **By making the template data-driven, the integrity of the template is maintained.**
- ✍ **A data-driven system can benefit from generic tools**
 - ✍ **For configuration**
 - ✍ **For log control and management**
 - ✍ **For testing and basic interrogation**



... Conclusion

- ✍ Things such as middleware and data management are commonly maintained
- ✍ Anybody in the team can diagnose another's system
- ✍ The systems are automatically documented by their definitions
- ✍ A fully operational server can be built with minimal code **VERY QUICKLY!**
- ✍ It works!
 - ✍ **6 systems already successfully use BISCOTo**