# **EVIDENCE – A control system for laboratory** applications and small-scale experiments

Oliver Grimm – ETH Zürich, Institute for Particle Physics, Schafmattstrasse 20, 8093 Zürich, Switzerland – oliver.grimm@phys.ethz.ch

## **Overview**

Experiment control systems (ECS) typically comprise several individual programs that

- require configuration information,
- produce data,
- need to exchange information,
- visualize data of other programs and monitor system health.

### EVIDENCE

ECS functions with	USES	DIM communication	lean and	controlized
minimum extra coding	established tools	layer	easy to adapt	Centralizeu

## **CERN's DIM**

### **Distributed Information Management**

Communication system for distributed/mixed environments Provides network transparent inter-process communication layer

C++, C, Fortran, Java and Python interfaces

Little extra coding required (work handled by DIM library)

Thread-based (data handlers internally executed sequentially)

Access control with standard tools (e.g. IPtables)

- Provide data, commands, procedure calls Server
- Subscribe to data Clients Send commands





*EVIDENCE* is designed for small-scale experiments for which established and comprehensive control systems (e.g. EPICS, DOOCS or PVS-II) are too complex.

The centralized approach defines the range of applications: for large systems, decentralization might be better.

## **EVIDENCE** Functionality

A small number of server programs (the ECS backbone) contain the basic functionality.

They use a standard C++ class suggested also for application programs: allows easy central system surveillance (message logging and error severity monitoring) and an efficient way to access configuration information.

DIM **Name Server** 

Central instance of DIM for communication between processes IP address needs to be known to all DIM servers and clients

### Configuration Server

Supplies configuration information to servers Configuration read from single text file (format .INI style) Clients informed automatically on changes to the file

# Sky Quality Monitor [SQM] sky.quality.com address = 4711 port = Period = 30

Central data, message and error logging

Make remote procedure calls Commands Notification if a server dies and automatic reconnection Individual TCP/IP connection to server

Provides service information and addresses Name server Involved transparently by DIM library Not involved once connection between server and client established

C. Gaspar, M. Dönszelmann and Ph. Charpentier, Computer Physics Communications **140**, 102 (2001) References: Web address http://dim.web.cern.ch/dim/

## **Graphical User Interface**

### **Evidence** Data Display

uses Qt and Qwt	display DIM	sending of generic	subscribes to DIM
for visualization	Services and histories	DIM commands	service only once

#### Qt and Qwt tool-kits available free of charge (LGPL)

Single subscription to DIM service limits server load if multiple widgets use same data

Easily portable: no operating-system specifics used, only standard C++ code, Qt/Qwt capabilities and DIM

GUI building follows standard Qt programming procedures.



#### Subscribes dynamically to all DIM services (except if excluded) and writes all updates to text file

References: Qt home page: http://qt.nokia.com/

#### Alarm **Monitor**

#### Monitors control system health

Checks severity of server message services and for existence of servers Generates master alarm (requiring explicit acknowledge to reset) Can send email if server has error or is down

#### History Server

Provides recent data (histories) of all DIM services Data kept in memory ring buffers for quick access Used mainly for user interface

Bridge

Transmits DIM data, commands and procedure calls between 2 DIM networks

Can limit network load on main servers (e.g. from multiple GUIs) Allows blocking of commands and procedure calls according to IP address

#### **Functions of Evidence class**

#### Starts DIM server

Provides textual status service SERVER\_NAME/Status Severity encoding (INFO, WARN, ERROR, FATAL)

Provides method for configuration request. If data not available and no default, terminates with FATAL message Allocated memory freed upon class destruction Requests over network only send in case configuration file changed

#### Server programming example

#include "Evidence.h"

// Class declaration class MyServer: public EvidenceServer { ... }

#### // Constructor

MyServer::MyServer(): EvidenceServer(SERVER\_NAME) { ... }

#### **GUI examples from FACT project First G-APD Cherenkov Telescope**

#### Evidence used for the first test module and initial camera commissioning

#### Main window

Access to all subsystems via detachable tabs 2 signal traces with statistics (mean and sigma) shown

Edd - Evidence Data Display - Node: ihp-pc1.ethz.ch:2505	×						
Menu							
Event scope 🔀 🛛 FADctrl 🗵 🛛 Bias 🗵 🛛 Feedback 🖾 🛛 FSC/SQM 🖾 🛛 Evidence 🖾							
Board Server started (Revision: 10143, compiled Jul 8 2011 07:45:39) 0 0							
4 Signals Run Header Event Header							
■ 4,0,4 (1269) - 0,0,0 (1349)	<b>-</b> 4,0,4 (1269) <b>-</b> 0,0,0 (1349)						
Channel 4.0.4 (1269): m -1.43, s 2.07 0,0.0 (1349): m -0.42, s 1.23 0,0.0 (1349): m -0.42, s 1.23							
-2 -2 -2 -2 -2 -2 -2 -2							
-10 -10 -10							
-15							
100 200 300 400 500							
Pixel display							

#### Service history window

Initial history obtained from History server (thus available on all GUI instances), subsequent accumulation by GUI

Stripchart mode keeps constant time span

Add traces by drag&drop or copy/paste

Extensive plot manipulations - zooming along all axes, panning - saving and printing



08:26 08:43 09:00 09:16 09:33 09:50

2-J (F2011 08:18:19 to 12-J (F2011 10:12:06

38 0625

Provides method for translating DIM service safely into text

Provides DIM exit and error handler

Installs signal handler SIGQUIT (Ctrl-Backspace), SIGTERM, SIGINT (Ctrl-C), and SIGHUP (terminal closed)

Catches un-handled C++ exceptions gcc-style termination handler

Allows browsing of data returned by History server

// Request configuration data char \*DataDir = GetConfig("datadir");

// Create service

int SizekB = 0; Service = new DimService(SERVER\_NAME "/SizekB", SizekB);

// Update all clients Service.updateService();

// Set warning message Message(WARN, "SizekB is %d", SizekB);

// Write something to the central log file SendToLog("Did you know SizekB was so large?"); Monitoring main *Evidence* servers

Alarm status or unavailability of alarm server announced by pop-up window, optional acoustic alarm



#### **Overview of 20 digitizer boards**

Show last day

Show last hou

Set update rate

Save as ASCII

Save plot Print plot

Plot help

Single trace

Clicking on widget brings up history Details of DIM service shown in bottom status bar when mouse over widget

0				Edd	d - FADctrl - Board 201	io 39			$\odot$ $\odot$ $\otimes$
10.0.130.128	512	1.9988	33.0625	36	36	34.25	0	58.7182	Connected
10.0.130.129	513	2.0009	33.6875	36.625	36.5625	34.625	0	36.5548	Connected
10.0.130.130	514	1.9988	33.75	38.375	36.6875	36.4375	0	59.1395	Connected
10.0.130.131	515	2.0009	35.1875	36.125	38.25	34.0625	0	33.7294	Connected
10.0.130.132	516	1.9988	33.625	36.3125	36.375	34.1875	0	74.5057	Connected
10.0.130.133	517	1.9988	33.5625	36.4375	36.375	34	0	74.164	Connected
10.0.130.134	518	2.0009	33.875	36.0625	36.5	34.3125	0	58.1142	Connected
10.0.130.135	519	1.9988	33.3125	37.9375	38.125	35.9375	0	72.4908	Connected
10.0.130.136	520	2.0009	34.3125	38.6875	36.375	36.375	0	43.8921	Connected
10.0.130.137	521	1.9988	33.125	37.0625	35.5	35.375	0	50.8239	Connected
10.0.131.128	768	2.0009	33.625	35.4375	36	33.9375	0	74.0083	Connected
10.0.131.129	769	1.9988	33.375	38.1875	36.375	35.625	0	33.7381	Connected
10.0.131.130	770	2.0009	33.5625	37.6875	37.9375	36.125	0	19.6423	Connected
10.0.131.131	771	1.9988	33.8125	36.3125	36.3125	34.0625	0	73.7898	Connected
10.0.131.132	772	2.0009	33.3125	35.625	35.3125	33.0625	0	50.7916	Connected
10.0.131.133	773	2.0009	33.625	36.125	36.25	33.8125	0	32.9694	Connected
10.0.131.134	774	1.9988	35.75	38.5625	38.1875	36.0625	0	50.8097	Connected
10.0.131.135	775	1.9988	33.4375	38.0625	36.1875	35.625	0	72.644	Connected
10.0.131.136	776	2.0009	33.125	35.6875	36.3125	34.0625	0	32.9259	Connected
10.0.131.137	777	2.0009	34.5	35.125	35.375	33.4375	0	32.9652	Connected

Acknowledgment: The help of Clara Gaspar (CERN) was indispensable for understanding the DIM system and making good use of it. EVIDENCE was developed within the FACT project (First G-APD Cherenkov Telescope)