CANONE – A HIGHLY-INTERACTIVE WEB-BASED CONTROL SYSTEM INTERFACE

M. Pelko, K. Zagar (Cosylab, Ljubljana), L. Zambon (ELETTRA, Basovizza, Trieste), A. J. Green (University of Cambridge, Cambridge)

Abstract

In the recent years, usability of web applications has significantly improved, approaching that of rich desktop applications. Example applications are numerous, e.g., many different web applications from Google. The enabling driver for these developments is the AJAX (Asynchronous JavaScript and XML) architecture. Canone, originally a PHP web interface for Tango control system developed at Elettra, is one of the first attempts of long-distance interaction with the control system via Web. Users with suitable privileges can create panels consisting of various graphical widgets for monitoring and control of the process variables of the control system on-line. Recently, Canone was extended to interact with a control system through an abstract DAL (Data Access Layer) interface, making it applicable to EPICS and TINE as well. Also, the latest release of Canone comes with drag'n'drop functionality for creating the panels, making the framework even easier to use. This article discusses the general issues of the web-based interaction with the control system such as security, usability, network traffic and scalability, and presents the approach taken by Canone.

INTRODUCTION

In the world of control system, the task of a web interface is to extend the environment of a classic control room to any PC all over the world without the necessity of installing any particular software.

The challenge of a web interface is to work in a hostile environment where security is the main concern and bandwidth limitation may be a severe obstacle.

Canone is a graphical animated web interface which can interact with most of the particle accelerator control systems. It was built as the web interface of the Tango control system [1] but with a well delimited communication protocol toward the control system. This part was later substituted by a more generic DAL (Data Access Library) [2] interface which supports EPICS [3], TINE [4], IFC etc.

Canone is built mainly in PHP (PHP Hypertext Processor) [5] and animated with AJAX [6] [7]. The main characteristic is given by the combination of a client side graphical library with the AJAX powered animation machine.

The external appearance is a set of "panels" which are composed of a certain number of "widgets" (graphical elements) and some HTML tags. Integrated development tools to edit panels and administer permissions online at runtime are included in the distribution.

*miha.pelko@cosylab.com

ARCHITECTURE AND CONNECTION TO THE CONTROL SYSTEM

A web interface is expected not to interfere heavily with the core of the control system for both security and performance reasons. On the other hand it should be highly interactive.

In order to achieve this Canone separates the web server (blue boxes in Figure 1) from the control system network. Web server is responsible for the interaction with the end client, handling graphical representations and animations. A Java based socket server (Gateway) acting as a single client on the control system network presents the connection between the control system network and the web server via TCP/IP protocol.

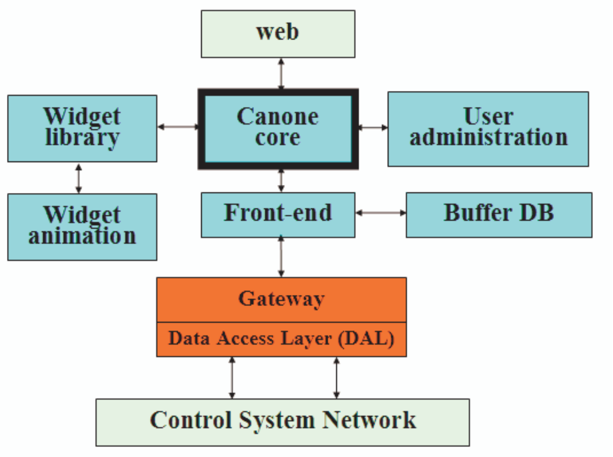


Figure 1: Block diagram of Canone

Web server acts as the client of the gateway, prompting the gateway when it requires status information (status request) or when it wants to issue a command to the control system (command request).

In order to decrease the total number of requests to the control system, the web server cashes the requests for a configurable amount of time in a database (SQLite or MySQL). If multiple clients of the Web server request the same information within this buffer time, only one request is issued to the gateway. With a similar mechanism multiple command requests of the same type are also filtered.

The installation requires a web server with little more then a basic LAMP (Linux Apache MySQL PHP) installation and a machine with the access to the control system network capable of running a simple Java application (gateway). The client side requires only a web browser with JavaScript and pop-ups enabled. Firefox is preferred browser but tests have been done with Internet Explorer, Opera and Safari.

Canone is originally a solution for controlling Tango from a web browser. Since January 2007 it can connect to any of the control systems trough the use of Data Access Library (DAL). This was achieved by developing a Java based socket server (gateway) using DAL to connect to the control system.

DAL prescribes a consistent set of Java interfaces for access to the control system's dynamic and configuration data. At this moment there are three control system implementation of DAL developed (EPICS, TINE and IFC).

As Canone requires the list of all the devices and variables available on the control system and as DAL API does not provide this (e.g. EPICS control system does not provide such functionality), this was solved trough the introduction of the ASCII file, listing the devices and variables.

GRAPHICS

Canone allows for building and customizing control panels. A simple panel is composed of a title and a set of widgets, but the page deployment may be as complex as allowed by HTML. Tables and external links are examples of features that can be added.

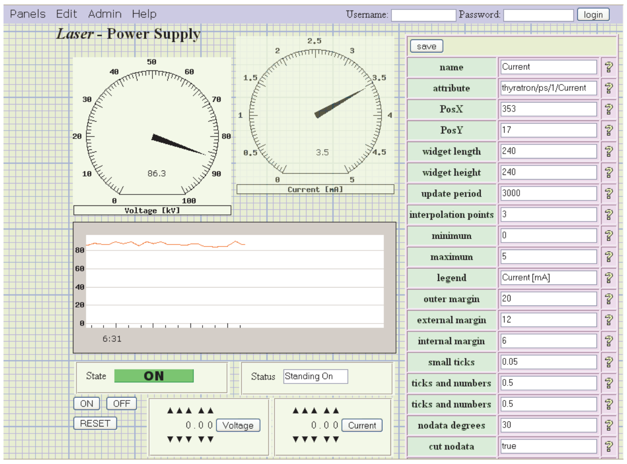
A panel can be built writing an independent PHP script which utilizes some of the basic Canone libraries (widget, front-end, user administration, etc.). Alternatively, a panel can be easily created and modified using a Canone integrated tool and then saved as a XML file on the server side (Figure 2).

```
<canone version='3.0.0 alpha'>
<panel name="laser/ps/1">
  <widget class='label' name="title">
    <content>Laser Power Supply</content>
    <x>120</x>
    <y>5</y>
  </widget>
  <widget class='analog' name="Voltage">
    <source>thyratron/ps/1/Voltage</source>
```

Figure 2: Fragment of panel description file.

A specific panel editing tool enables to move or resize each widget with a drag-and-drop technique. In order to get an adequate precision the normal background is substituted with a scaled grid image, the widget that is being moved or resized is alpha-blended and all coordinate change can be seen in real time in the configuration table on the right side of the graphic editor (Figure 3).

Depending on the browser some offset in coordinates might be present. In this case the coordinates reported in the configuration table are different from the real coordinates by a constant factor; this problem is easily fixed using the incorporated “Edit offsets” menu option.



name	Current
attribute	thyratron/ps/1/Current
PosX	353
PosY	17
widget length	240
widget height	240
update period	3000
interpolation points	3
minimum	0
maximum	5
legend	Current [mA]
outer margin	20
external margin	12
internal margin	6
small ticks	0.05
ticks and numbers	0.5
ticks and numbers	0.5
nodata degrees	30
cut nodata	true

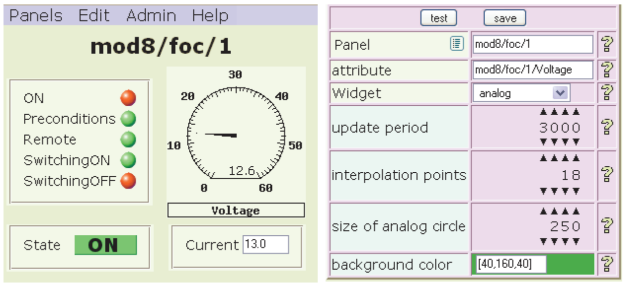
Figure 3: Drag-and-drop graphic editor

Widgets

A widget is a graphic element representing either a variable or a command.

Each widget is contained in its own class which extends the generic "widget" class. A fundamental attribute of this class is called "config" and contains all the parameters that can be customized. For each parameter, a data structure is composed by an identifier, a type definition, the default value, a short description and a long description (intended for the on line help). The *setParam()* method allows to set all the parameters with a single string. Some widgets are implemented as PNG images and others as tables. In both cases the HTML code necessary to visualize the widget is returned by the *plot()* method.

Each widget of a given panel is associated to a control system variable or command. A form allows a customisation of all the widget parameters (Figure 4) including colours, by means of a selection pop-up. The bandwidth utilized to transfer the widget initialization has been significantly reduced utilizing JSON (JavaScript Object Notation).



Panel	mod8/foc/1
attribute	mod8/foc/1/Voltage
Widget	analog
update period	3000
interpolation points	18
size of analog circle	250
background color	[40,160,40]

Figure 4: an example of Canone panel and widget configuration table.

ANIMATION AND AJAX

A basic characteristic of a widget is how often it is refreshed. For most widgets only a small part has to be changed, as all the graduated scales and graphical ornaments can remain in background. By transmitting only the numeric data and not transmitting all the graphical elements at each refresh, efficiency is highly improved. AJAX is a framework which allows the asynchronous transmission of tiny packets of data in a highly efficient way.

JavaScript allows the dynamic update of only a fraction of a web page while all the rest remains unchanged. This is easy for all widget whose animated part is in text format, but it is more complex for the fully graphical widgets. The adopted solution borrows a generic library by Walter Zorn [8] which utilizes the background colour of floating DIVs, a HTML feature supported by all modern browsers. Despite a very smart optimization has been done to improve the efficiency of the graphical library, it is important to limit the number of animated graphical elements.

All background elements are built on the server side by a PHP routine, sent as an image to the client and never refreshed.

On the client side, the browser only depicts the minimum graphical elements right over the background image through the AJAX mechanism (Figure 5).

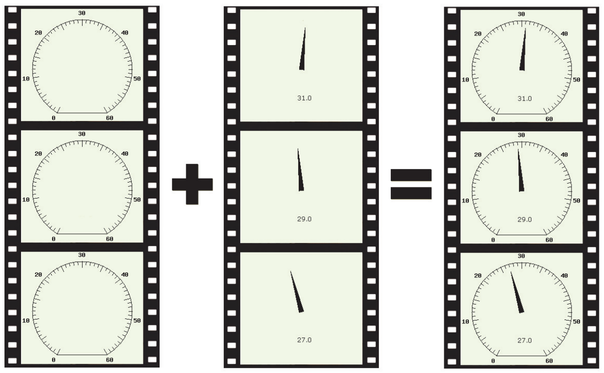


Figure 5: Widget animation.

AJAX is also asynchronous, which is very useful whenever a command is sent (i.e. a button is pressed). Meanwhile variables can continue being updated on the page. All this is resolved with a tiny delay and no other negative side effects such as commands lost or mishandled.

To save further bandwidth and improve the animation vividness, a configurable number of interpolated values between two consecutive readings can be inserted. In this way, a panel can have a refresh rate of up to 20 frames per second. The price is a delay in the response.

User Administration

The user authentication process in a web environment must be robust and cannot assume the loyalty of users at any time.

In Canone, the access is validated using the information contained in a database (SQLite or MySQL). Users must be registered in the system with *username* and *password*. Each user must be part of a user group, permissions are granted to user groups. In addition, several IP numbers (and net-masks) may be associated to a user group and receive the same rights. The rights granted to a user through its *username* and *password* override the ones associated to the IP number.

Three levels of permissions which can be set are:

- *read*: only read operations are granted, any write operation is denied
- *operator*: both read and write operations are granted
- *expert*: both read and write operations are granted, panels can be modified.

A special group called "admin" has permissions to create and delete users and grant or revoke permissions.

The user administration utility is composed of three tools:

- *access control:* grants permissions to user groups over panels and controlled devices
- *users management*: creates, searches, modifies and deletes user accounts
- *database*: is a generic database graphical client to change any configuration. Only expert administrators should use it.

CONCLUSION

The aim of Canone is to compete with any non-web graphic user interface. By now it already encompasses most of the functionality of traditional interfaces, sometimes even surpasses them (e.g. by including user management system, runtime drag'n'drop panel generation and customization, …).

The final goal is to make all the traditional GUIs completely obsolete. Although some important features are still missing the goal seems to be reachable in few years. The next important step can be to add a reliable support to alarms.

The original source code, demos and images can be downloaded from www.elettra.trieste.it/~tango/Canone.

REFERENCES

[1] L. Zambon, M. Lonza, "Web GUIs for the Tango control system", PCaPAC2006, Newport News, October 2006
[2] I. Kriznar et al., "Beyond Abeans", ICALEPCS, Knoxwille, October 2007
[3] http://www.aps.anl.gov/epics/
[4] Piotr Bartkiewicz *et al*, "TINE as an accelerator control system at DESY" *Meas. Sci. Technol.* **18** 2379-2386, 2007
[5] http://www.php.net
[6] M. Mahemoff, "AJAX Design Patterns", O'Reilly, June 2006
[7] http://www.modernmethod.com/sajax/
[8] http://www.walterzorn.com